Tutorial T1
Fundamentals of Music Processing:
An Introduction using Python and Jupyter Notebooks

Audio Structure Analysis

Meinard Müller, Frank Zalkow

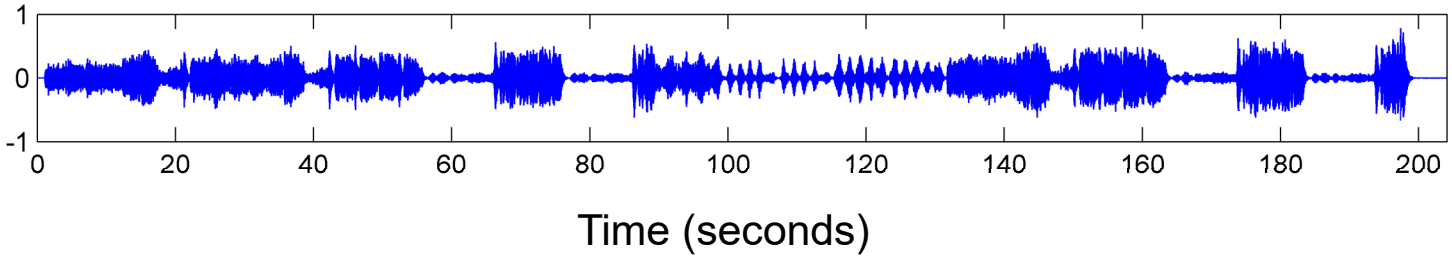
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de, frank.zalkow@audiolabs-erlangen.de

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Fraunhofer
IIS

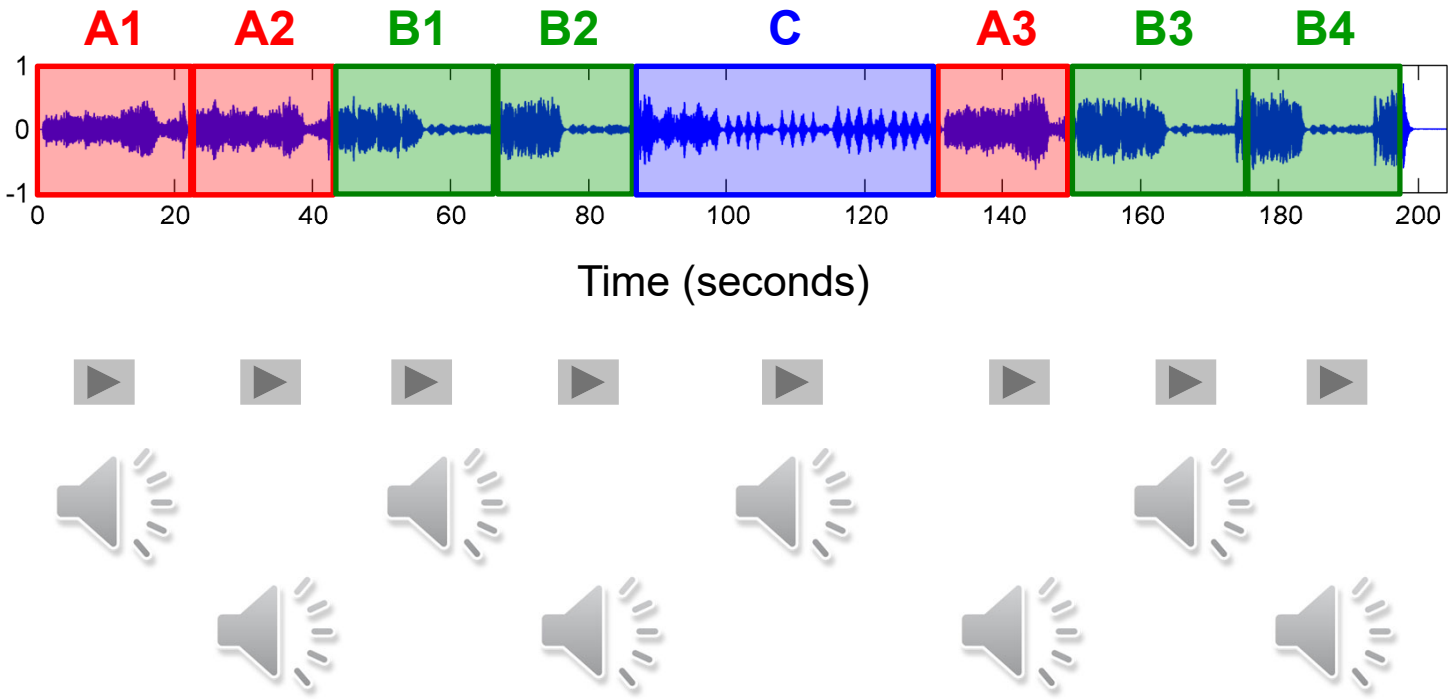
Music Structure Analysis

Example: Brahms Hungarian Dance No. 5 (Ormandy)



Music Structure Analysis

Example: Brahms Hungarian Dance No. 5 (Ormandy)



Music Structure Analysis

General goal: Divide an audio recording into temporal segments corresponding to musical parts and group these segments into musically meaningful categories.

Examples:

- Stanzas of a folk song
- Intro, verse, chorus, bridge, outro sections of a pop song
- Exposition, development, recapitulation, coda of a sonata
- Musical form ABACADA … of a rondo

Music Structure Analysis

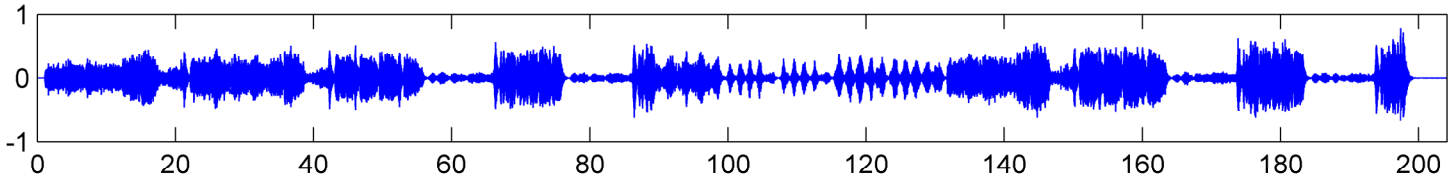
General goal: Divide an audio recording into temporal segments corresponding to musical parts and group these segments into musically meaningful categories.

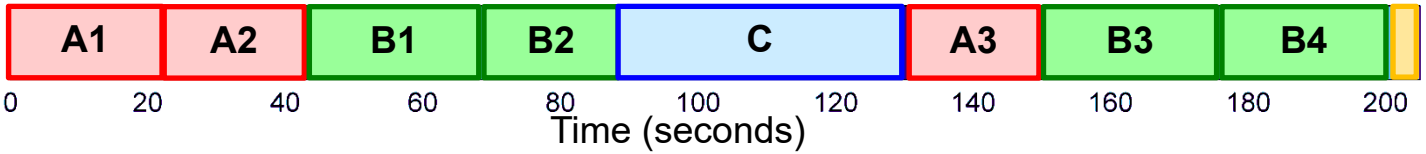
Challenge: There are many different principles for creating relationships that form the basis for the musical structure.

- Homogeneity: Consistency in tempo, instrumentation, key, …
- Novelty: Sudden changes, surprising elements …
- Repetition: Repeating themes, motives, rhythmic patterns,…

Feature Representation

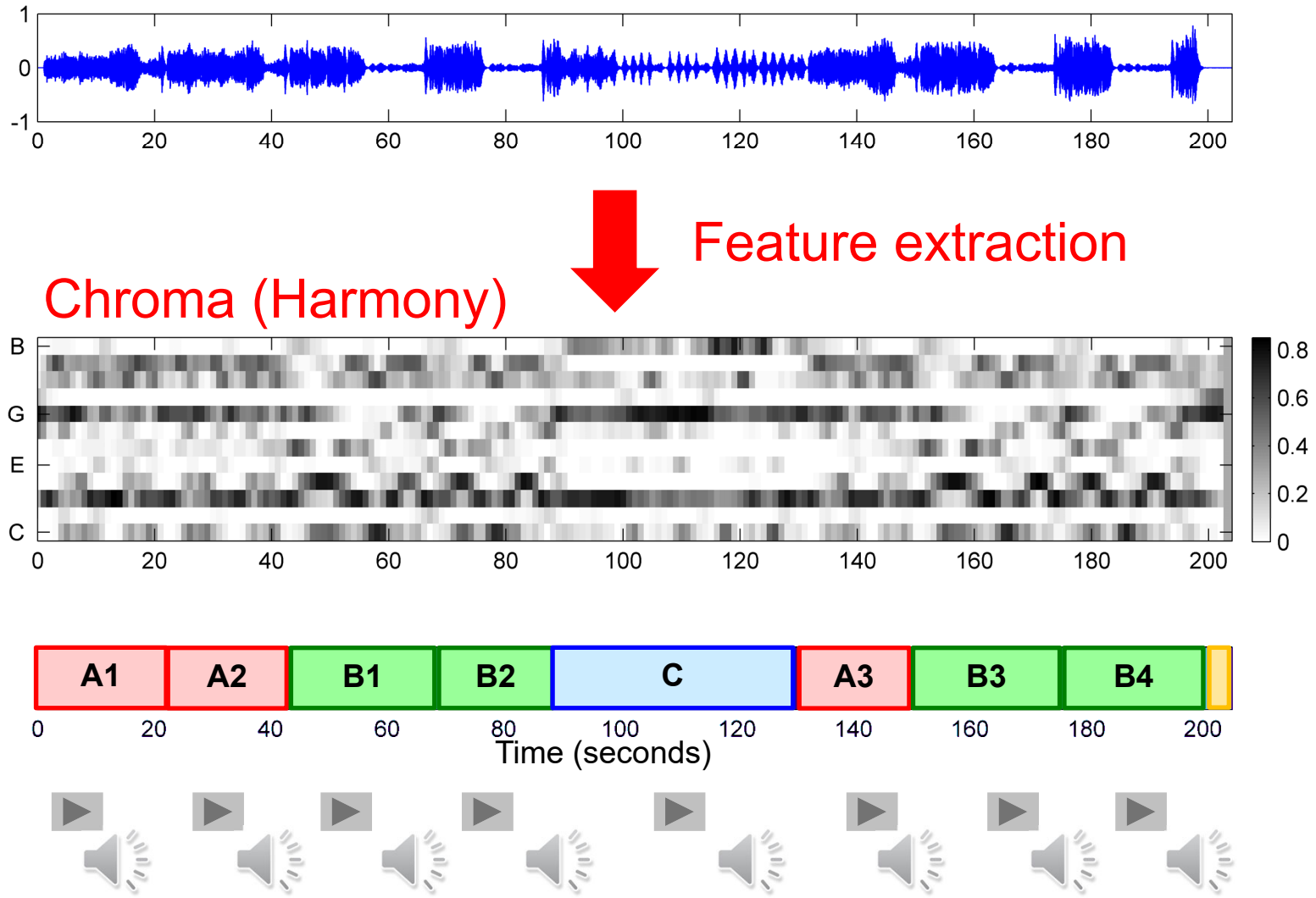
Example: Brahms Hungarian Dance No. 5 (Ormandy)





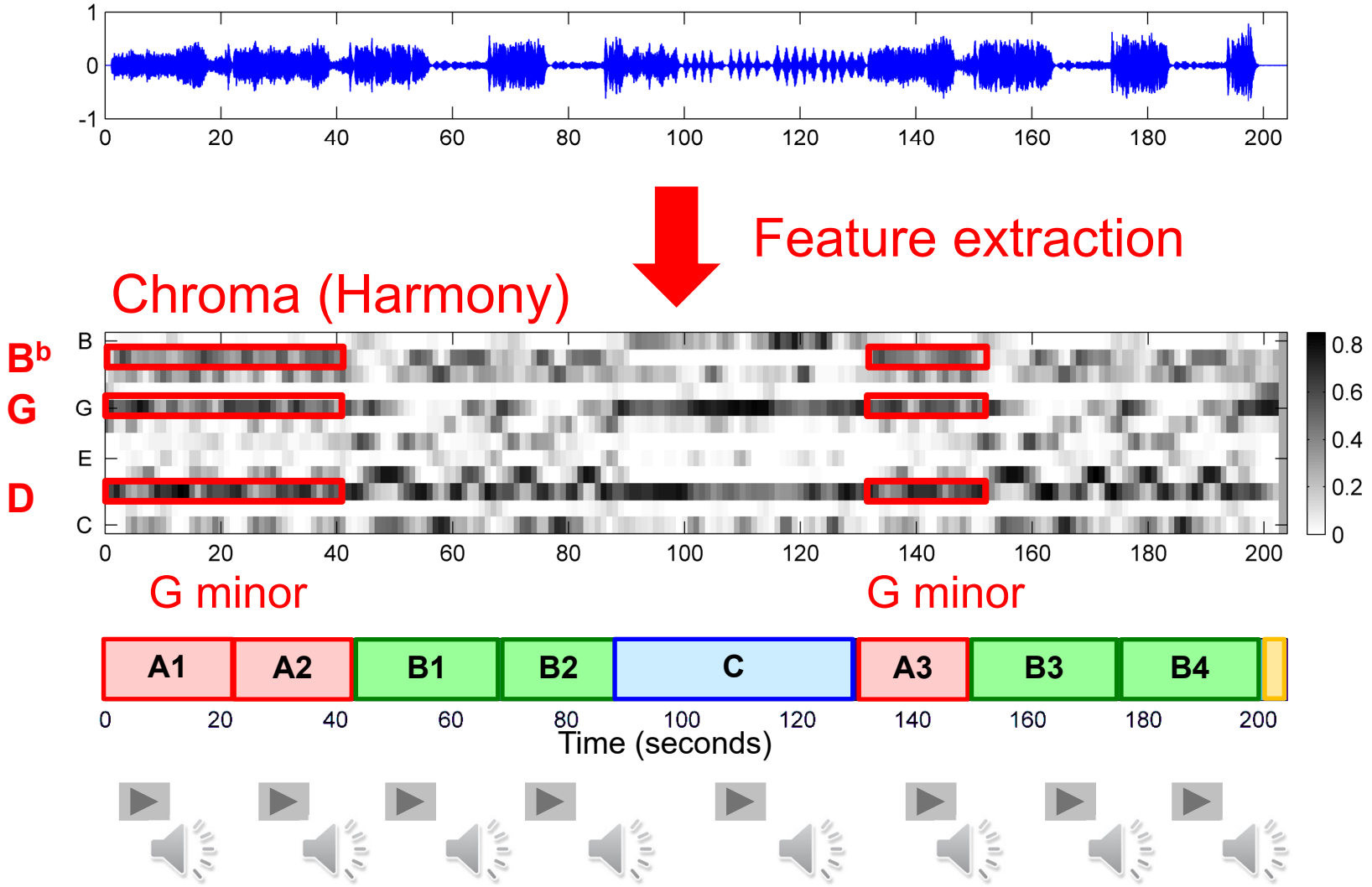
Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



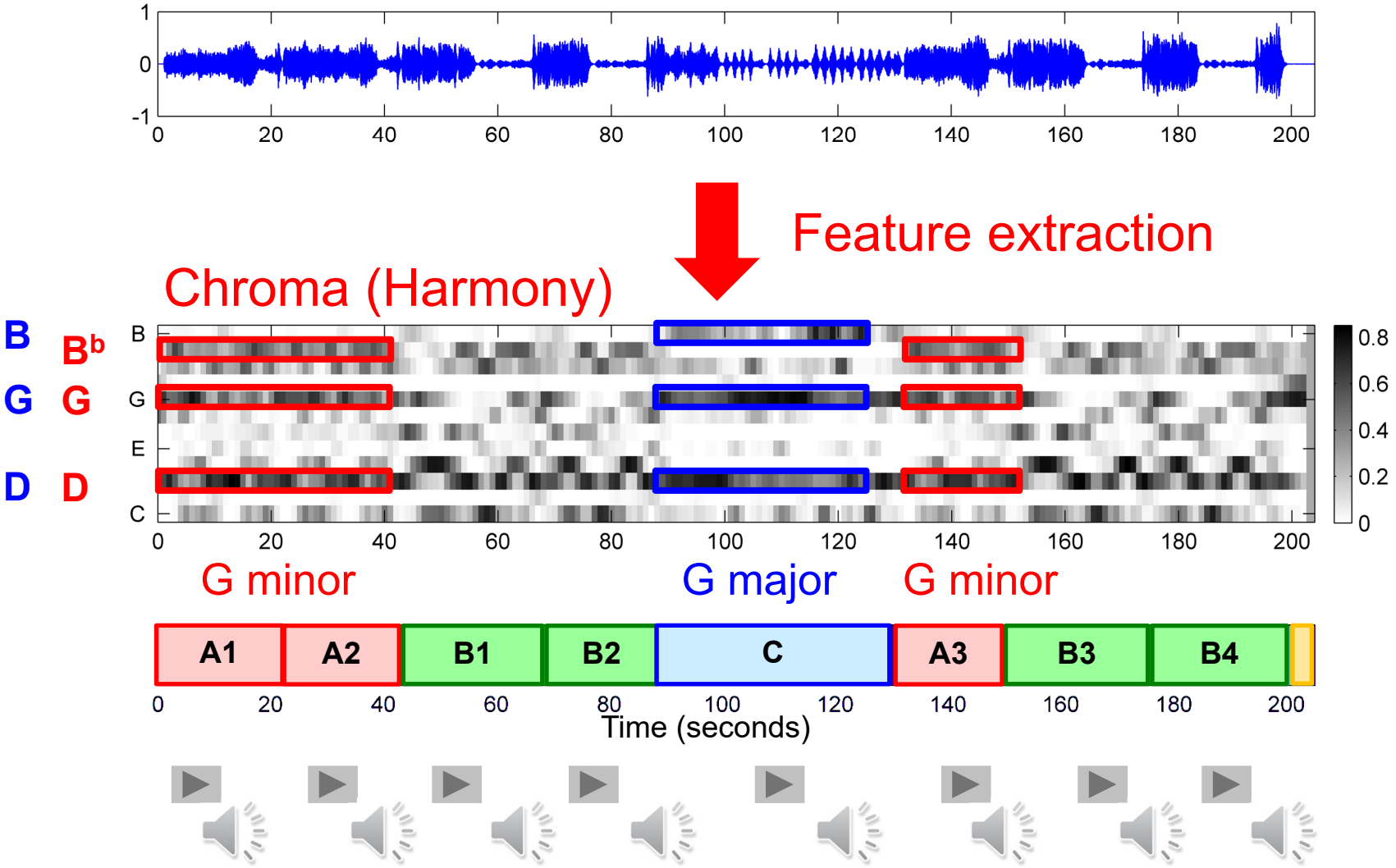
Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



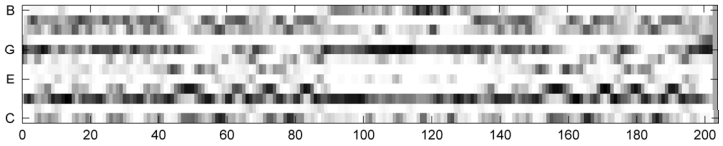
Self-Similarity Matrix (SSM)

General idea: Compare each element of the feature sequence with each other element of the feature sequence based on a suitable similarity measure.

→ Quadratic self-similarity matrix

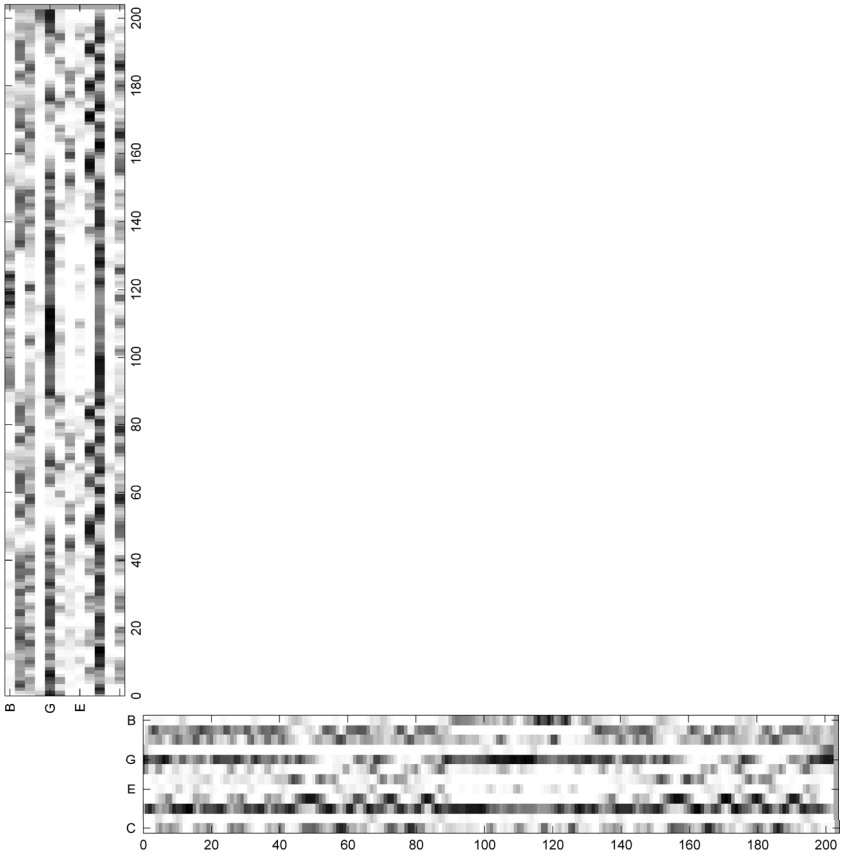
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



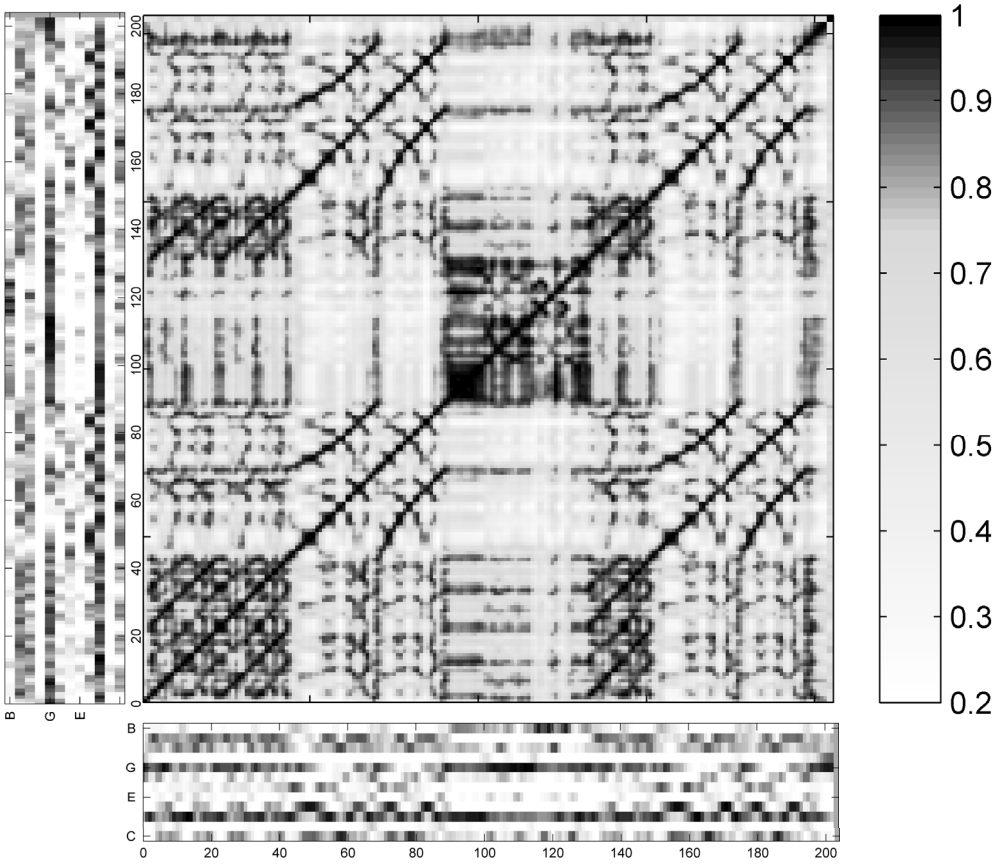
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



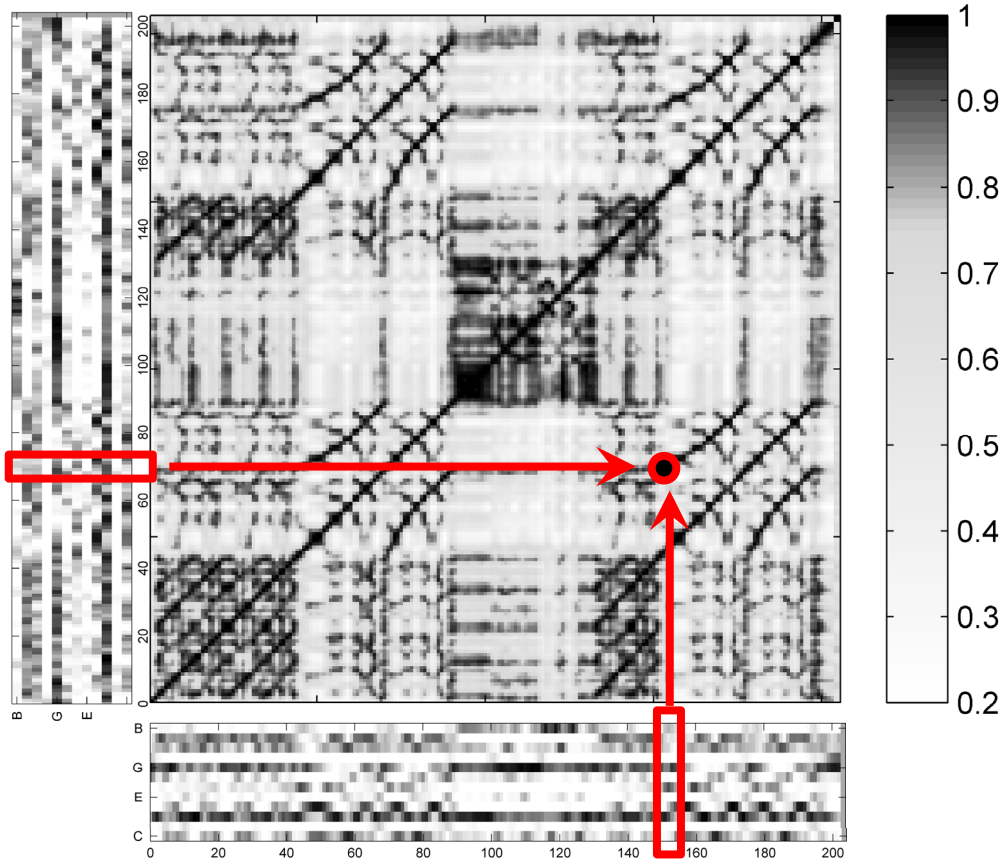
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



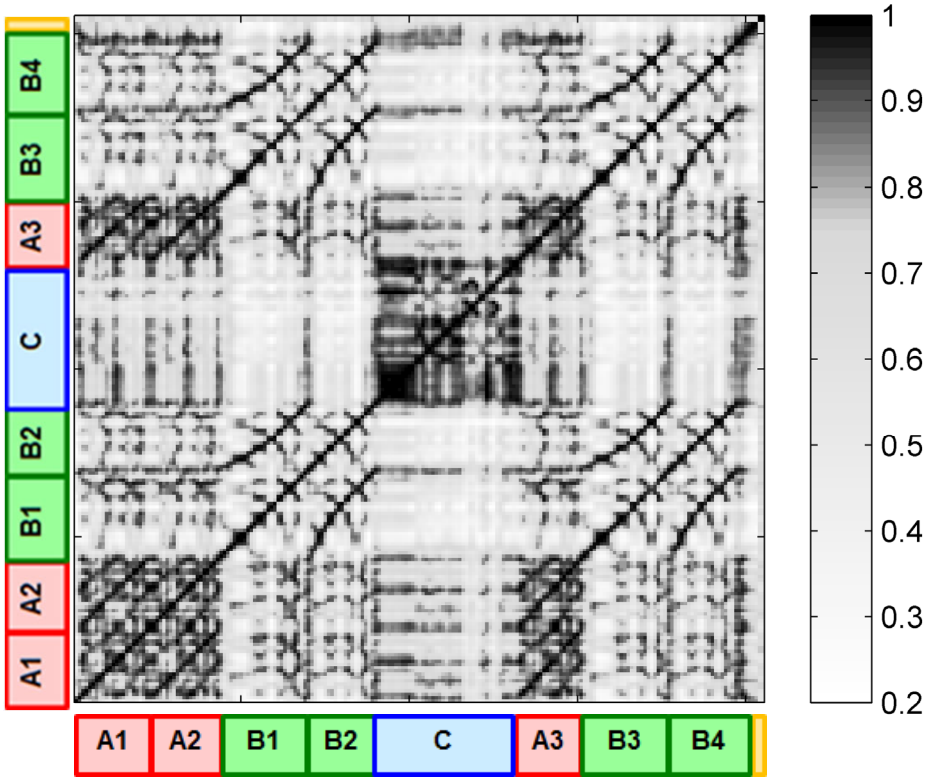
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



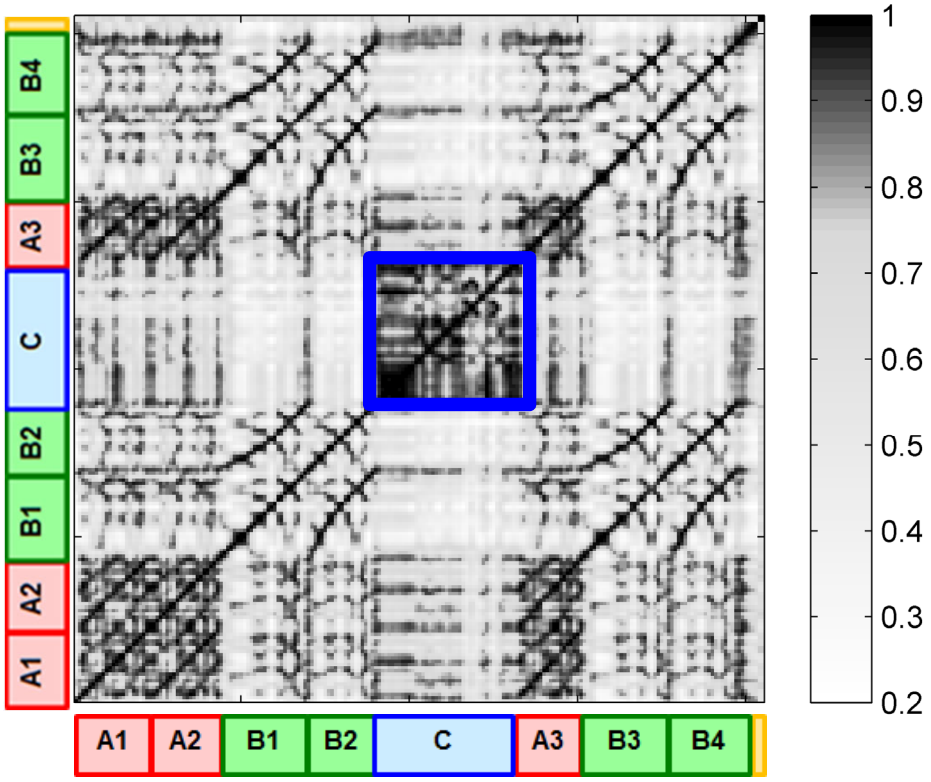
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



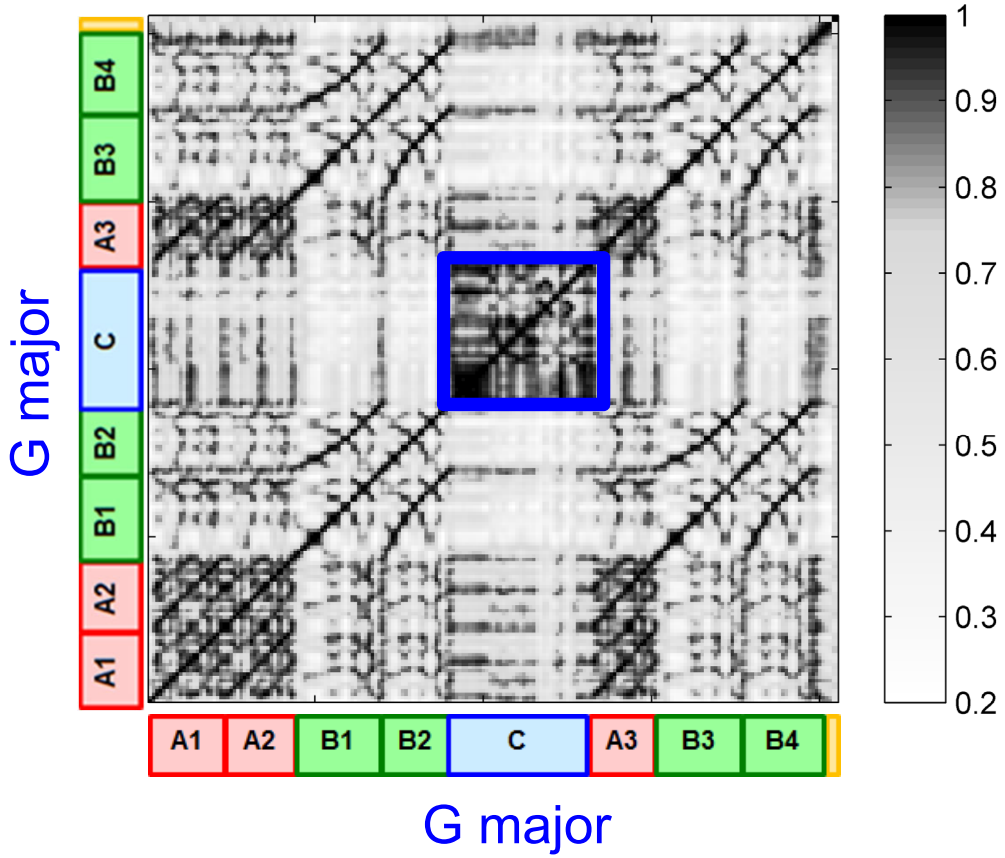
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



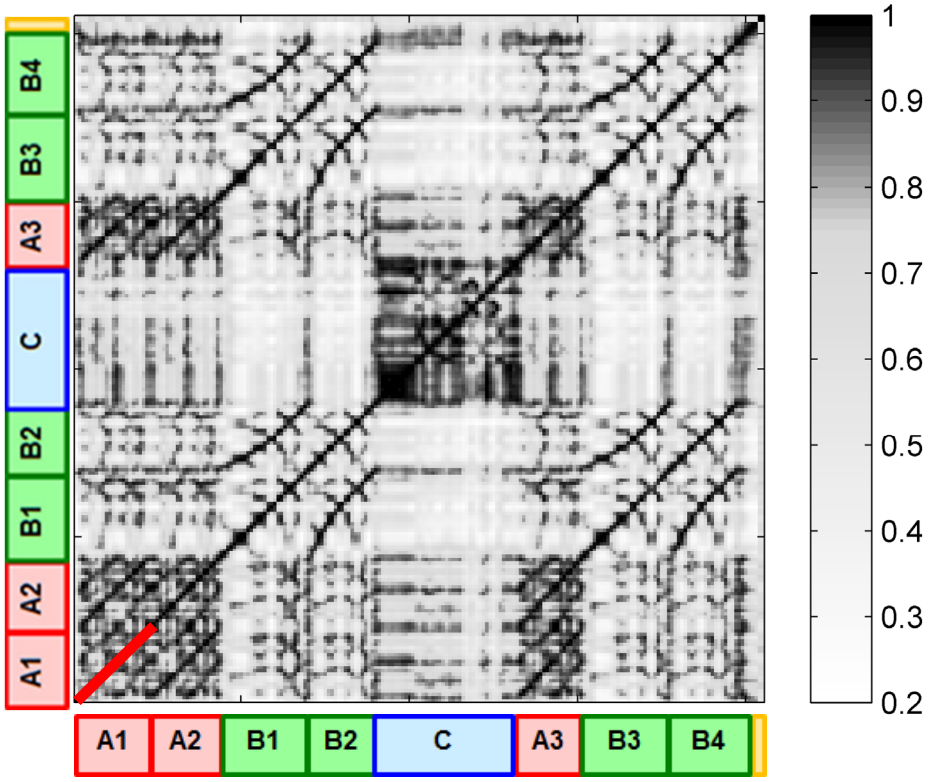
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



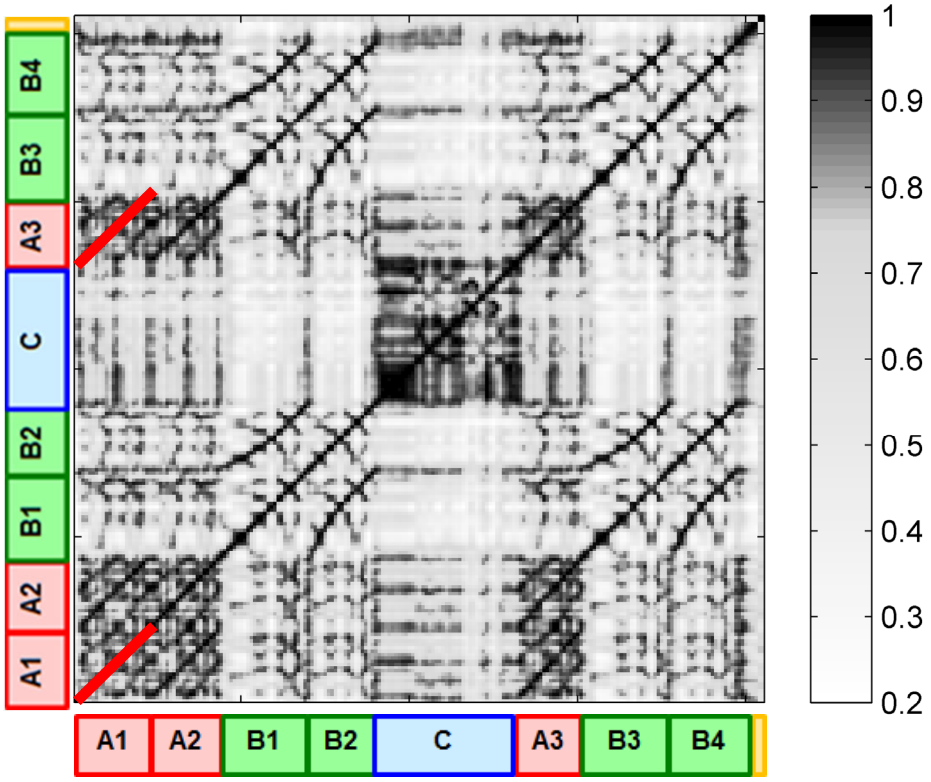
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



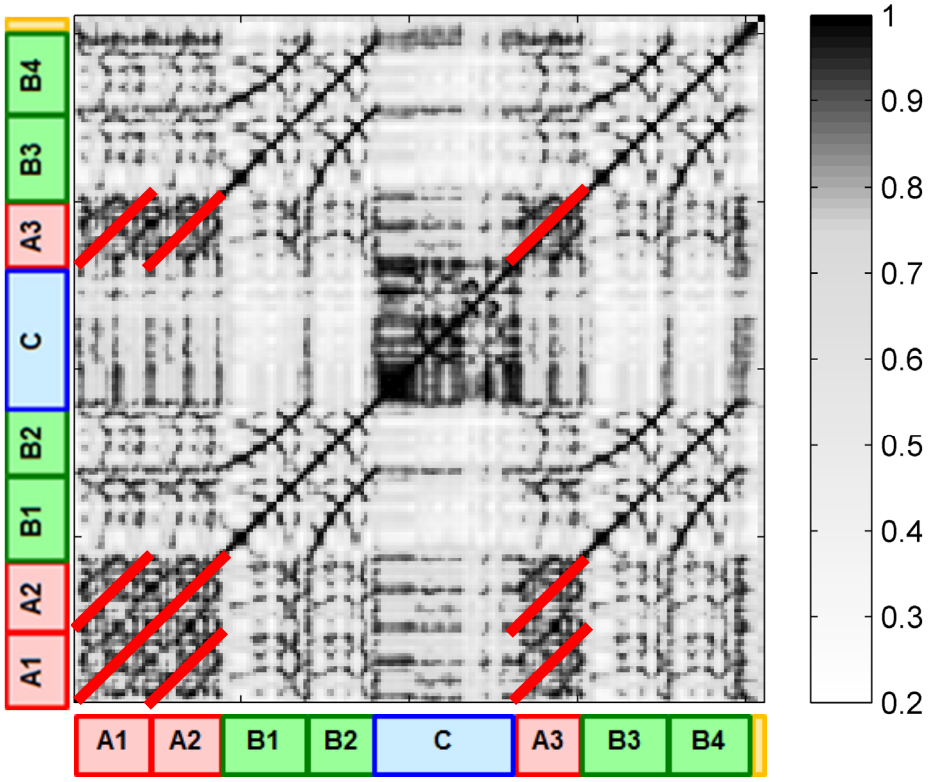
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



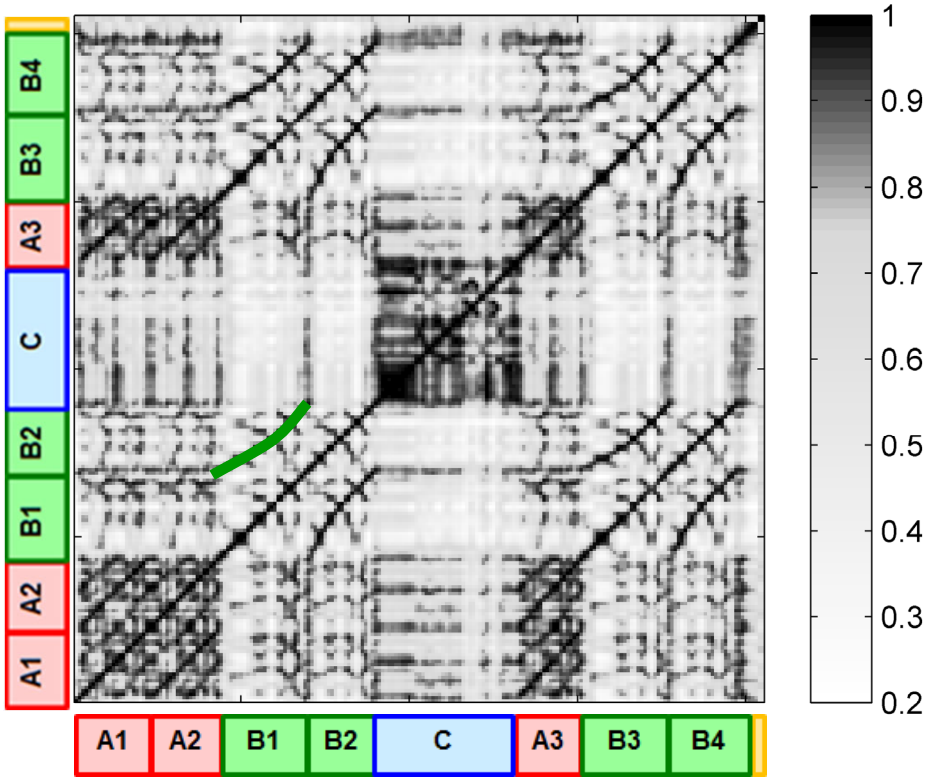
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



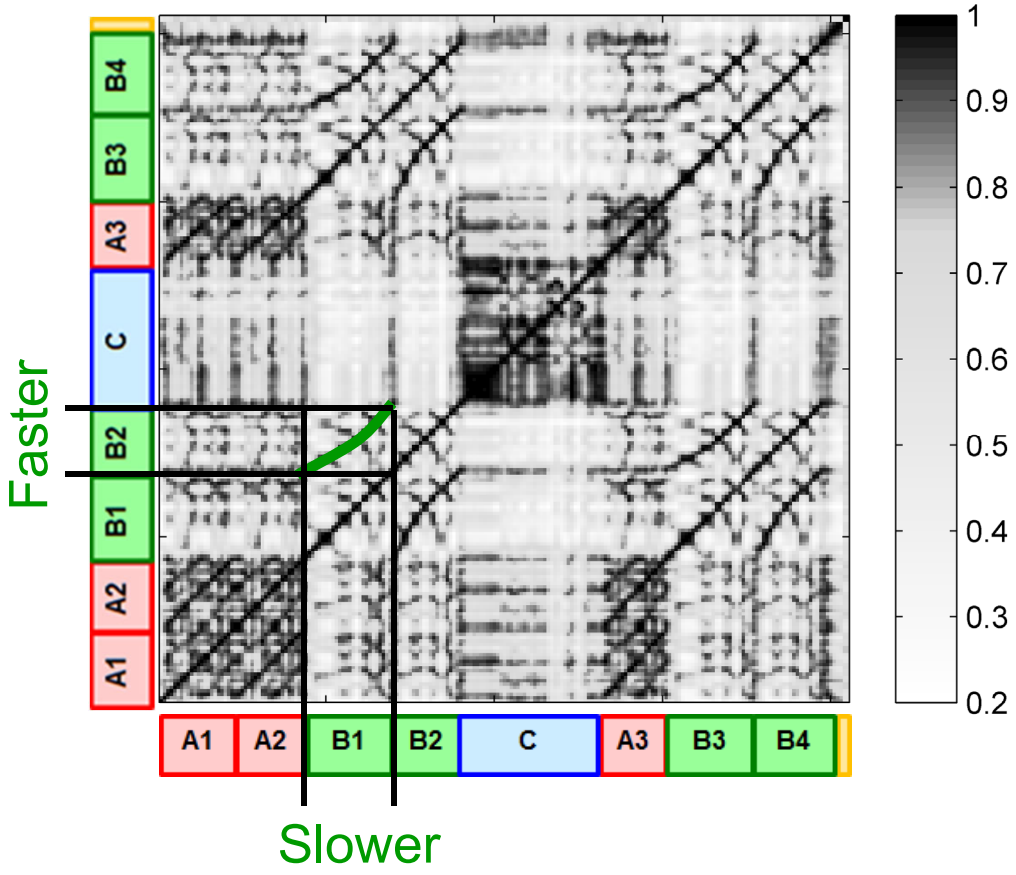
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



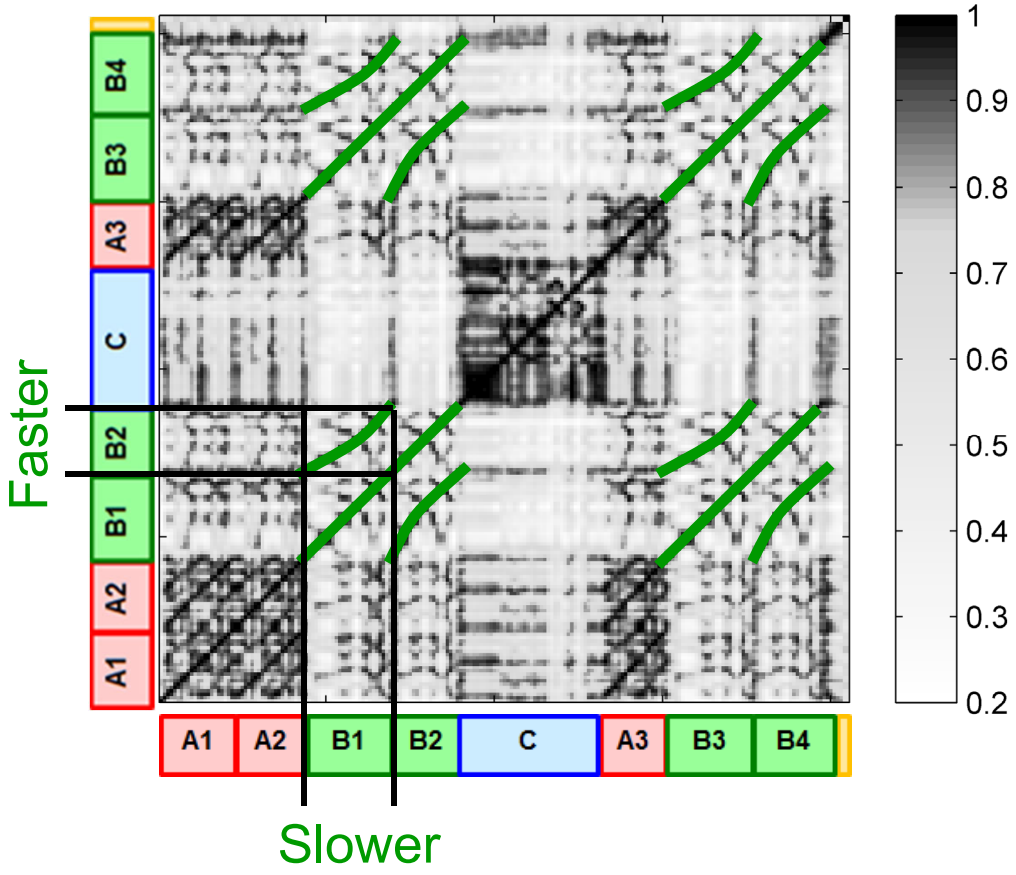
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



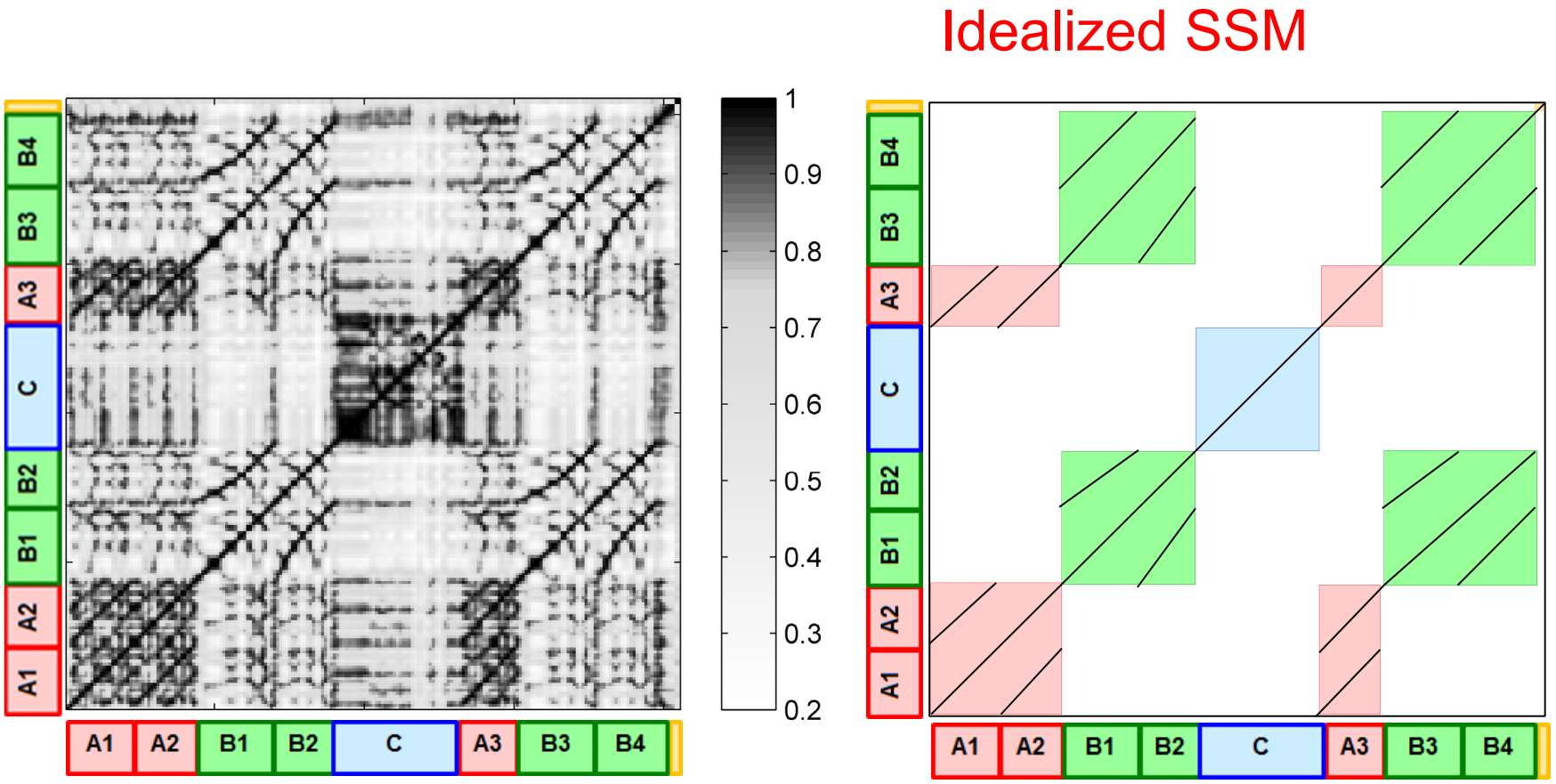
Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



Self-Similarity Matrix (SSM)

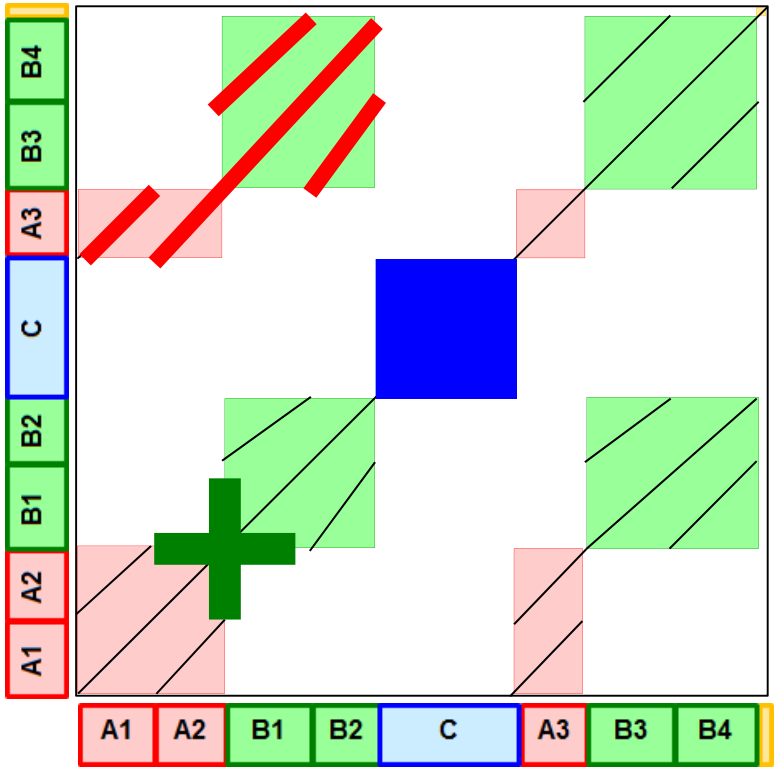
Example: Brahms Hungarian Dance No. 5 (Ormandy)

Blocks: Homogeneity

Paths: Repetition

Corners: Novelty

Idealized SSM



SSM Enhancement

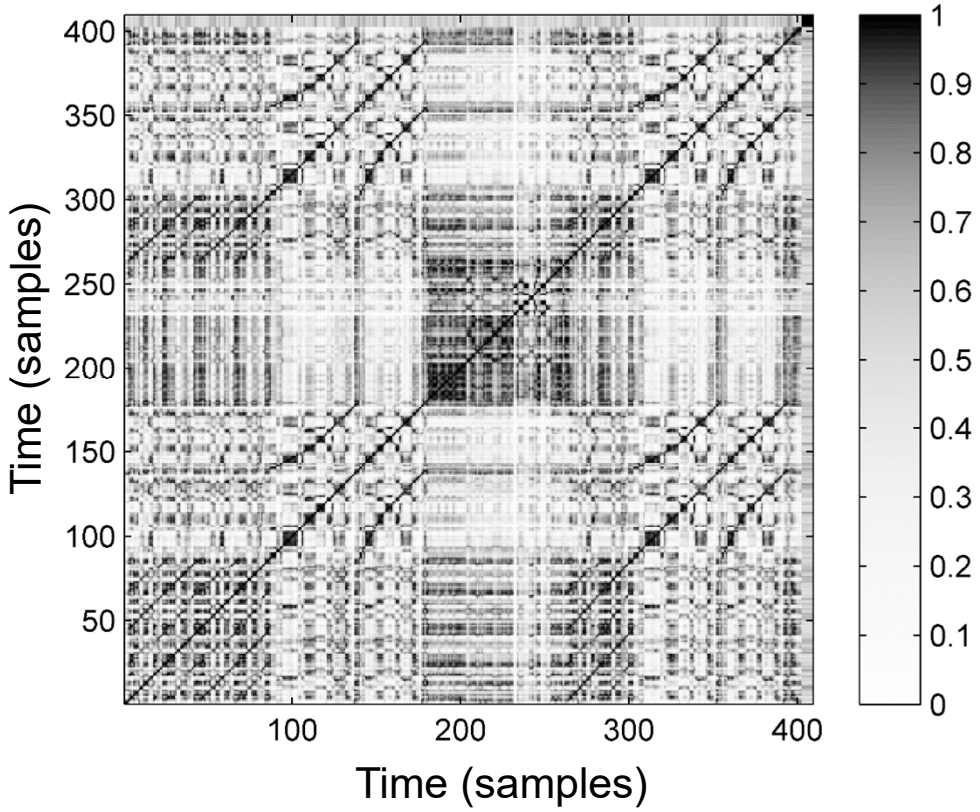
Challenge: Presence of musical variations

- Fragmented paths and gaps
- Paths of poor quality
- Regions of constant (low) cost
- Curved paths

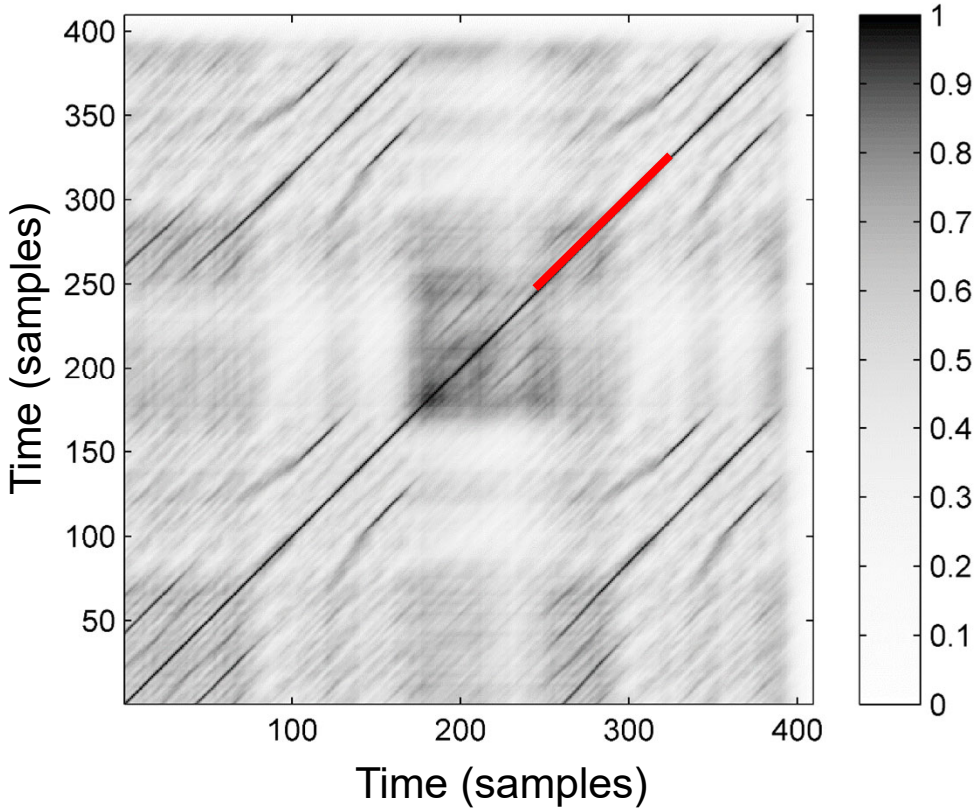
Idea: Enhancement of path structure

SSM Enhancement

Path Enhancement



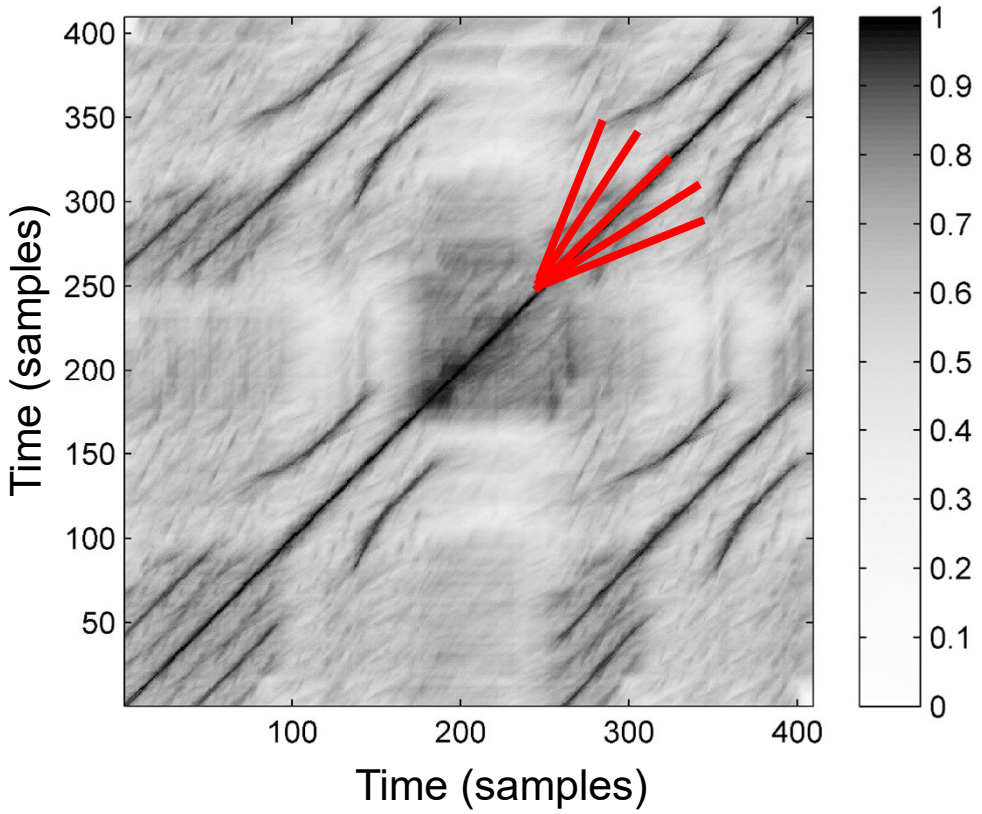
SSM Enhancement



Path Enhancement

- Diagonal smoothing

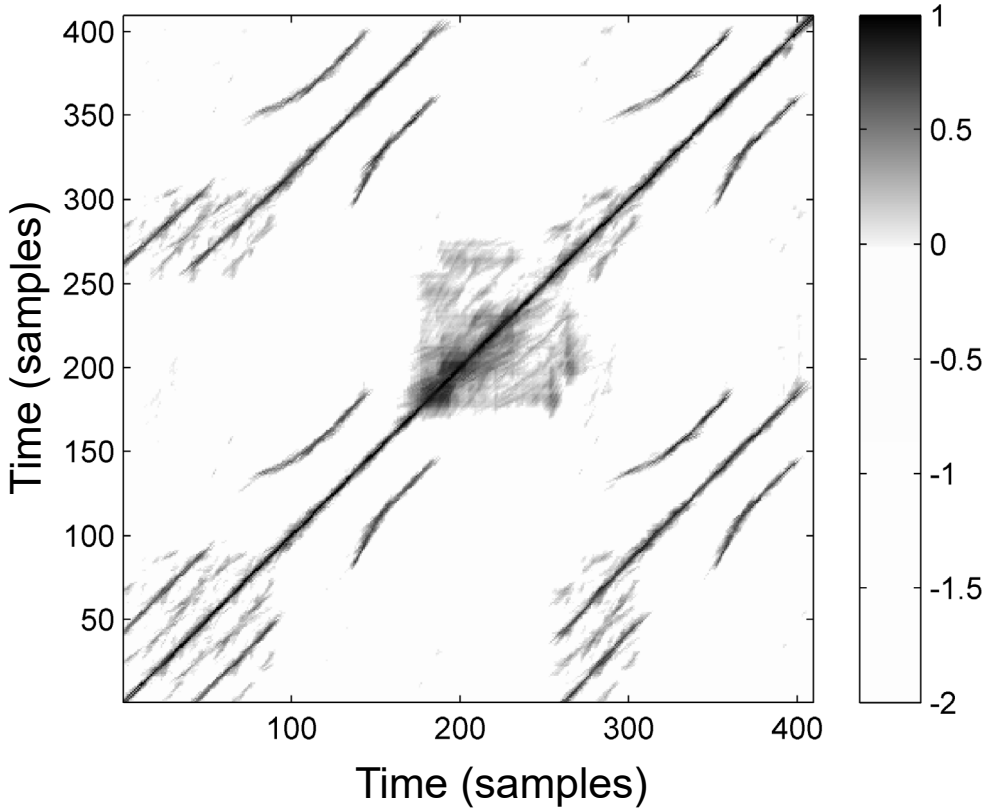
SSM Enhancement



Path Enhancement

- Diagonal smoothing
- Multiple filtering

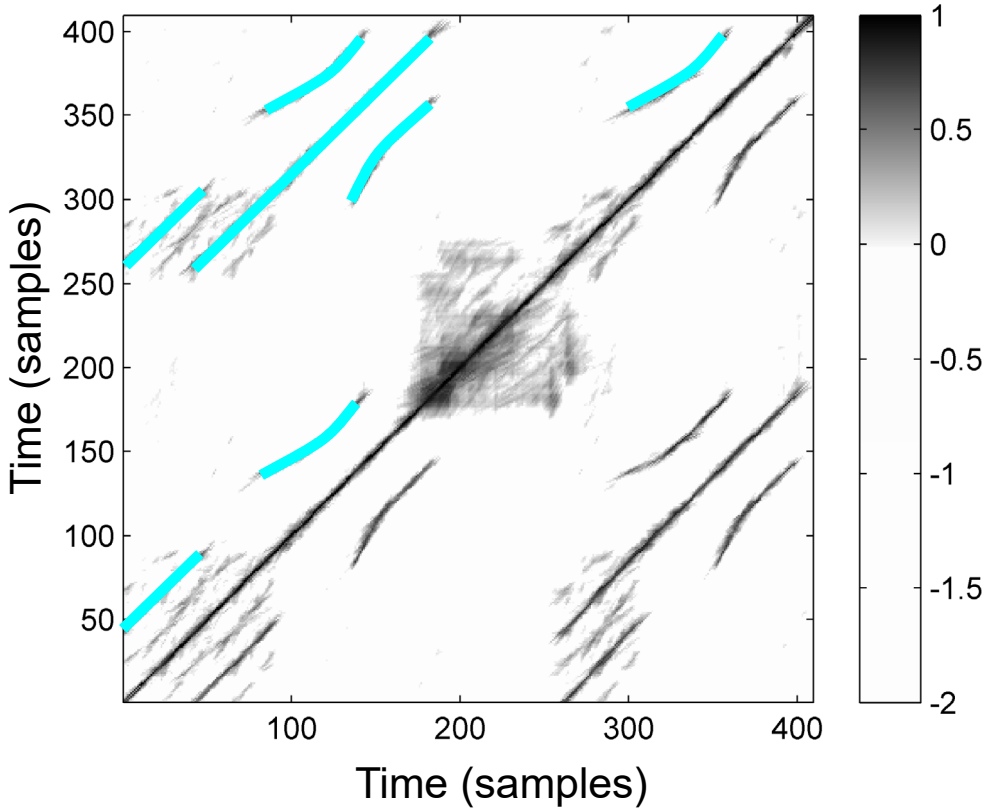
SSM Enhancement



Path Enhancement

- Diagonal smoothing
- Multiple filtering
- Thresholding (relative)
- Scaling & penalty

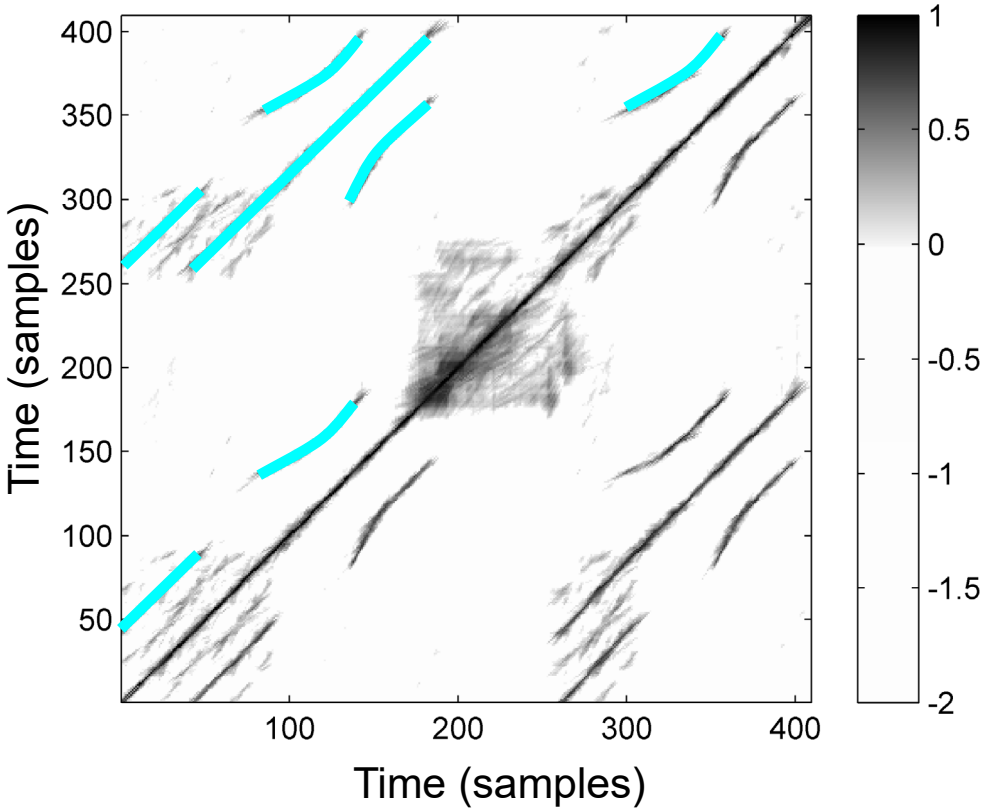
SSM Enhancement



Further Processing

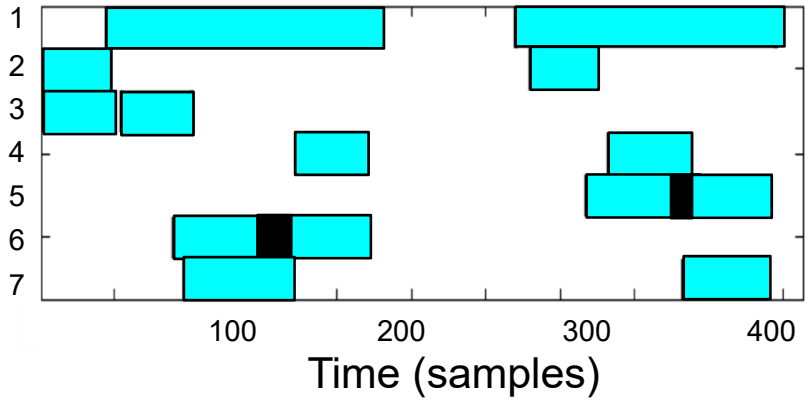
- Path extraction

SSM Enhancement

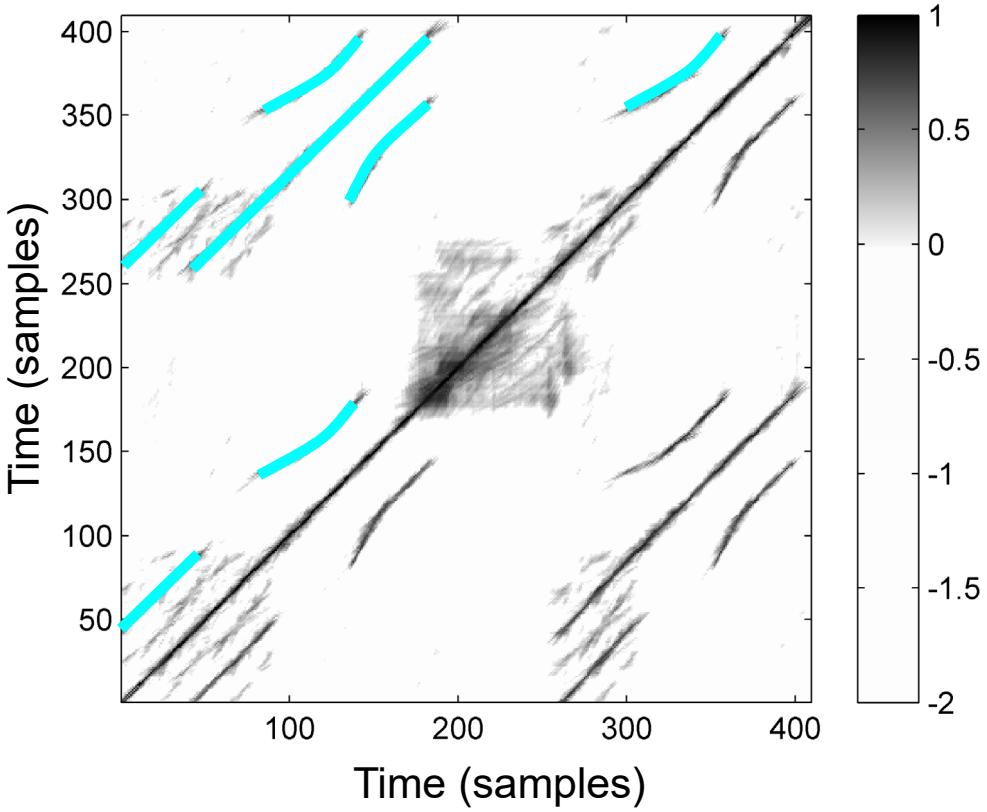


Further Processing

- Path extraction
- Pairwise relations

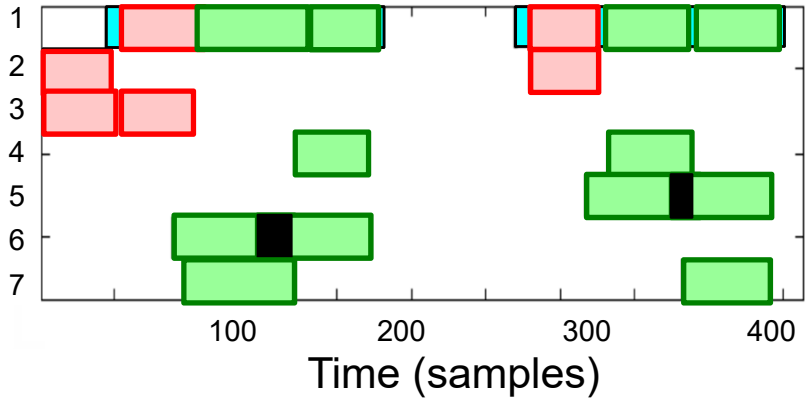


SSM Enhancement

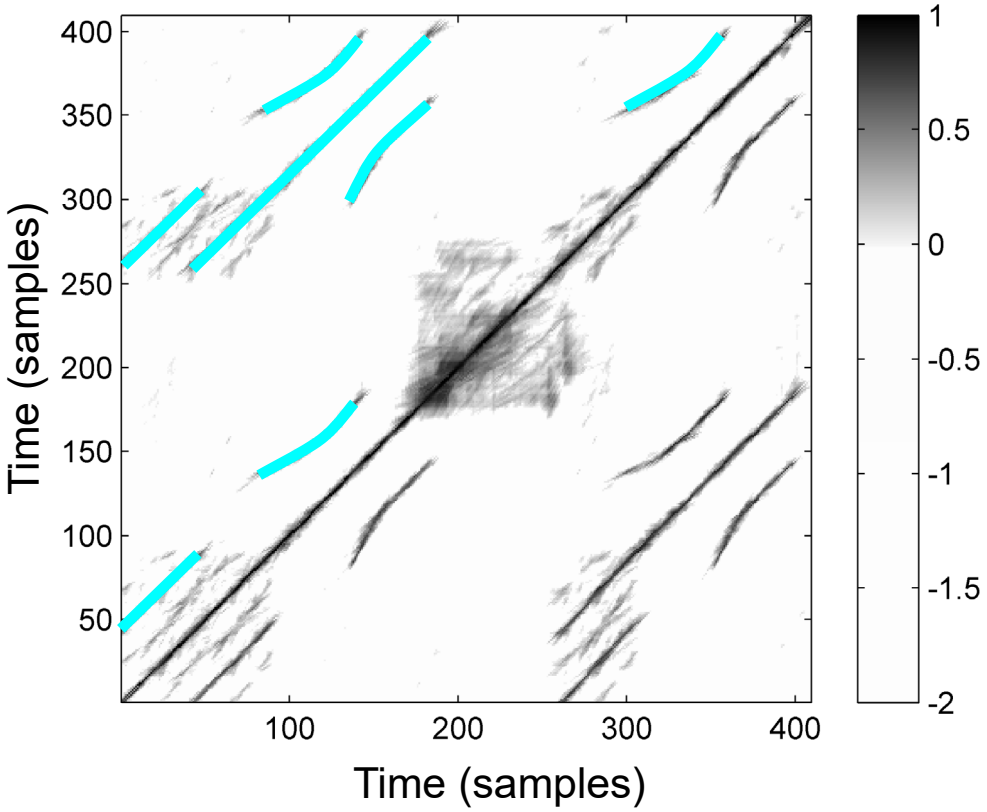


Further Processing

- Path extraction
- Pairwise relations
- Grouping (transitivity)

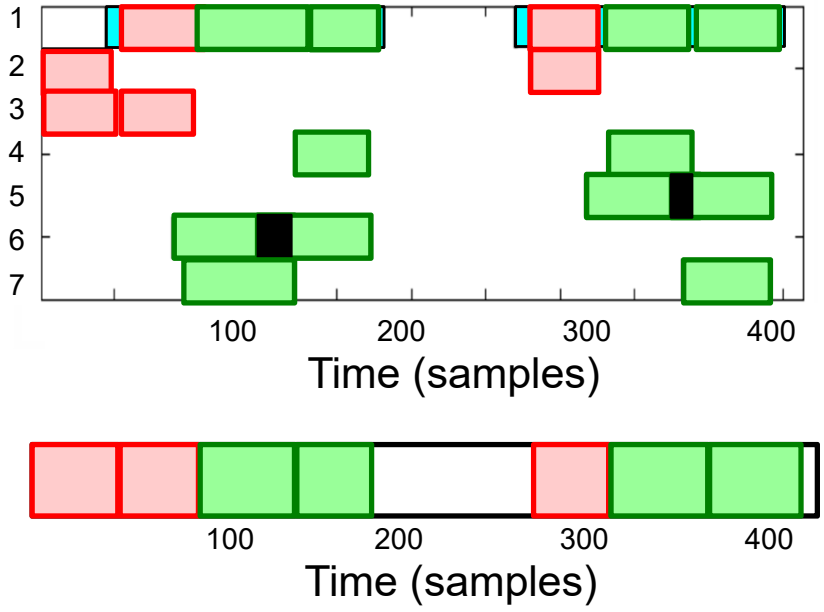


SSM Enhancement



Further Processing

- Path extraction
- Pairwise relations
- Grouping (transitivity)



Novelty-Based Segmentation

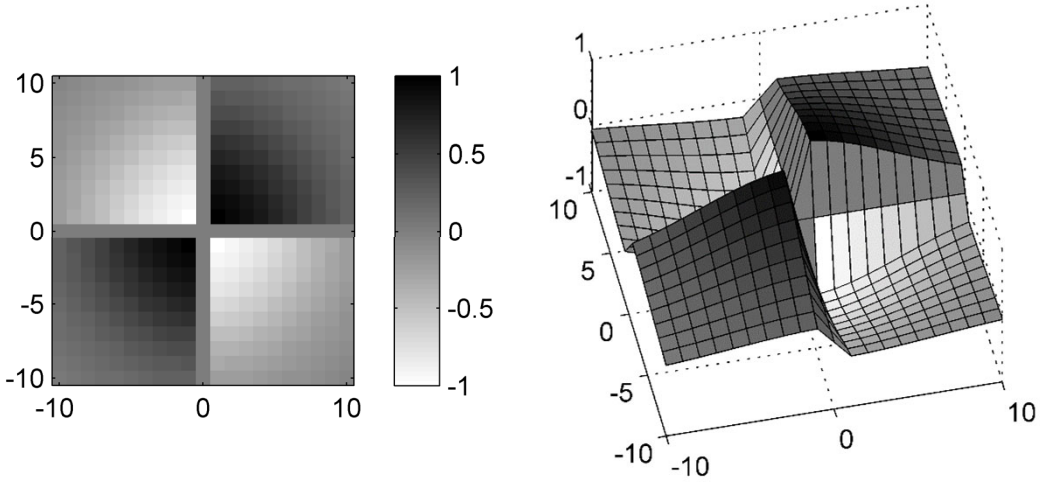
Novelty-Based Segmentation

General goals:

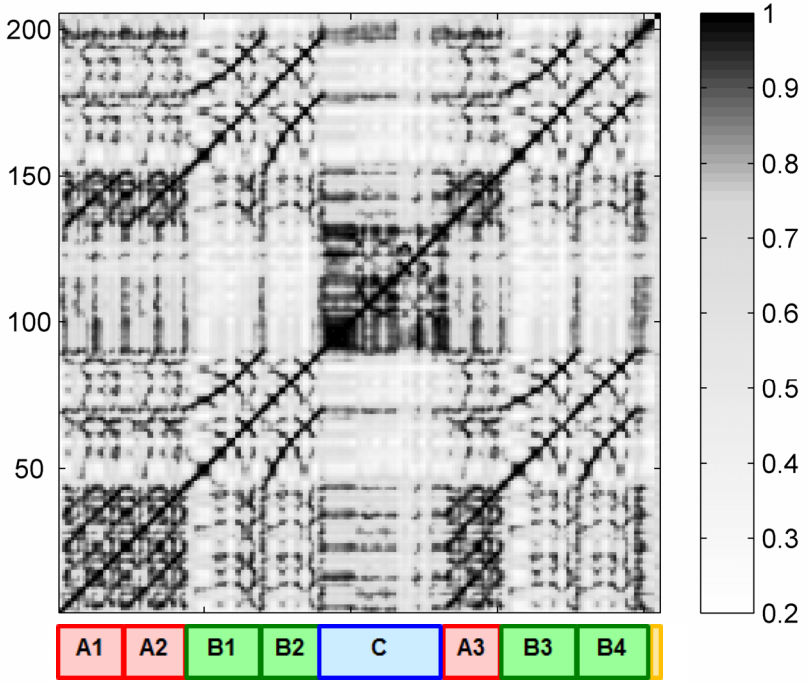
- Find instances where musical changes occur.
- Find transition between subsequent musical parts.

Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.



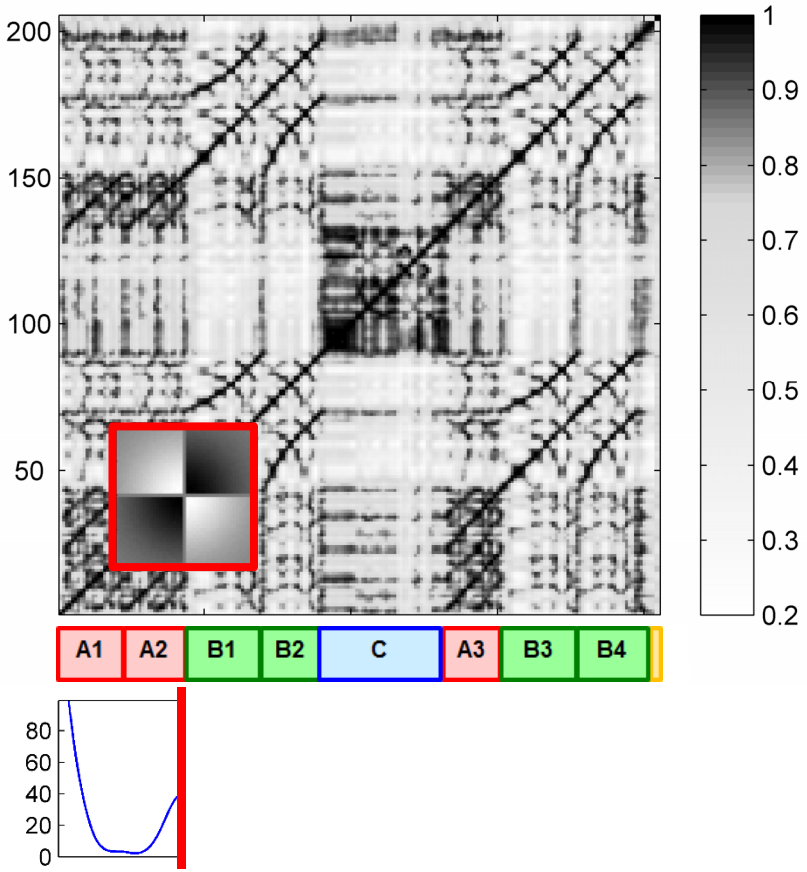
Novelty-Based Segmentation



Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

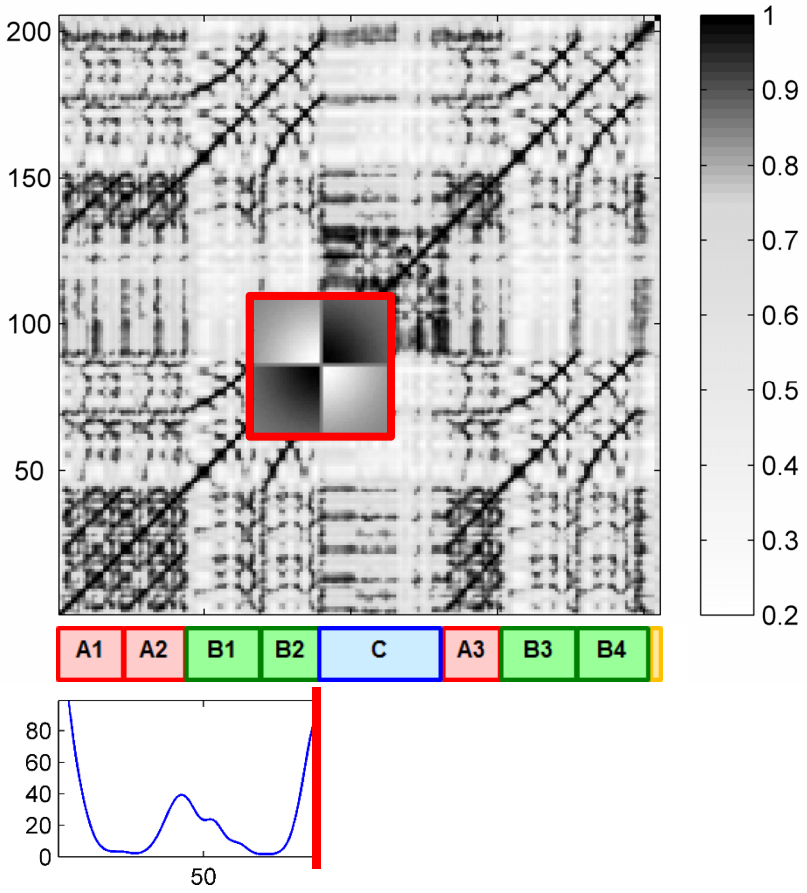
Novelty-Based Segmentation



Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

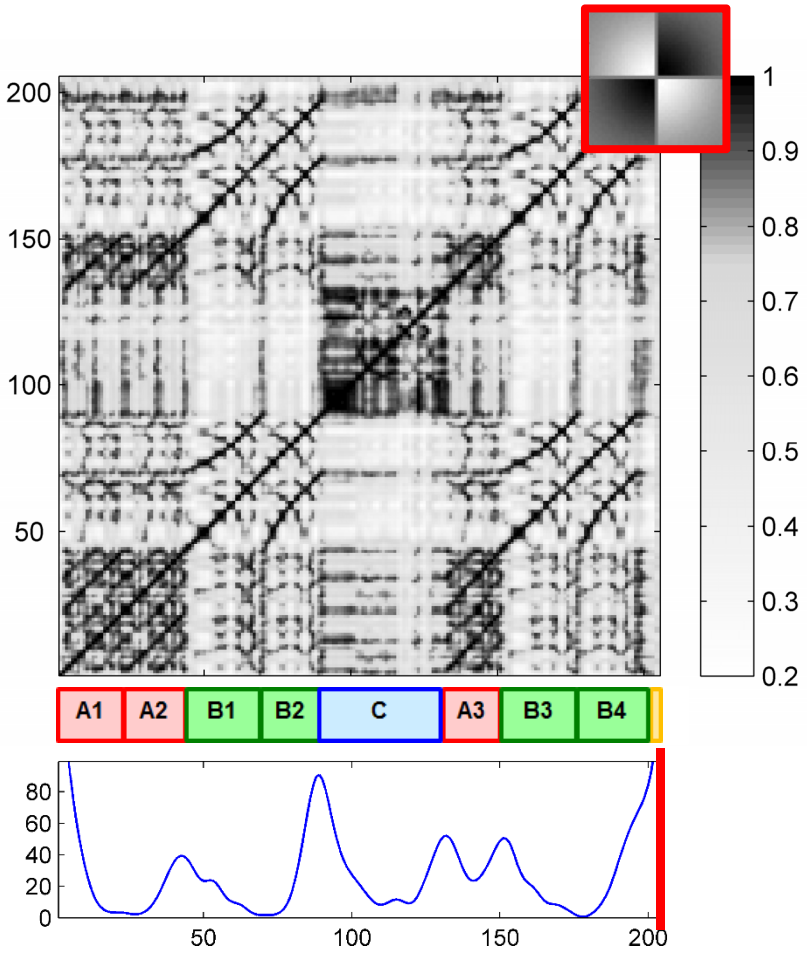
Novelty-Based Segmentation



Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

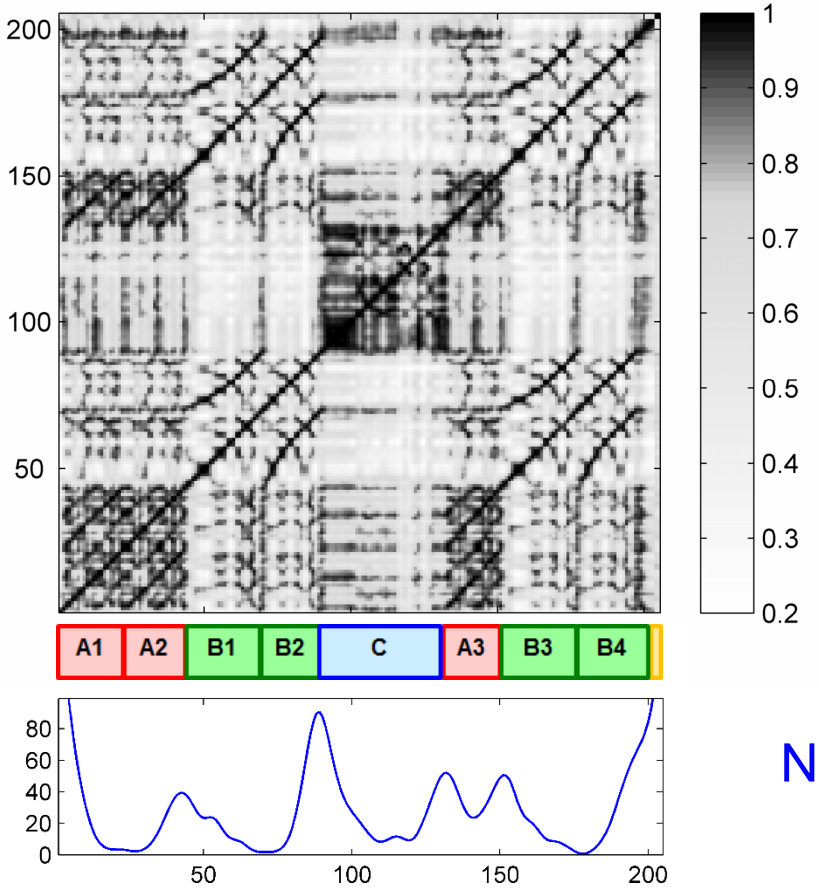
Novelty-Based Segmentation



Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

Novelty-Based Segmentation

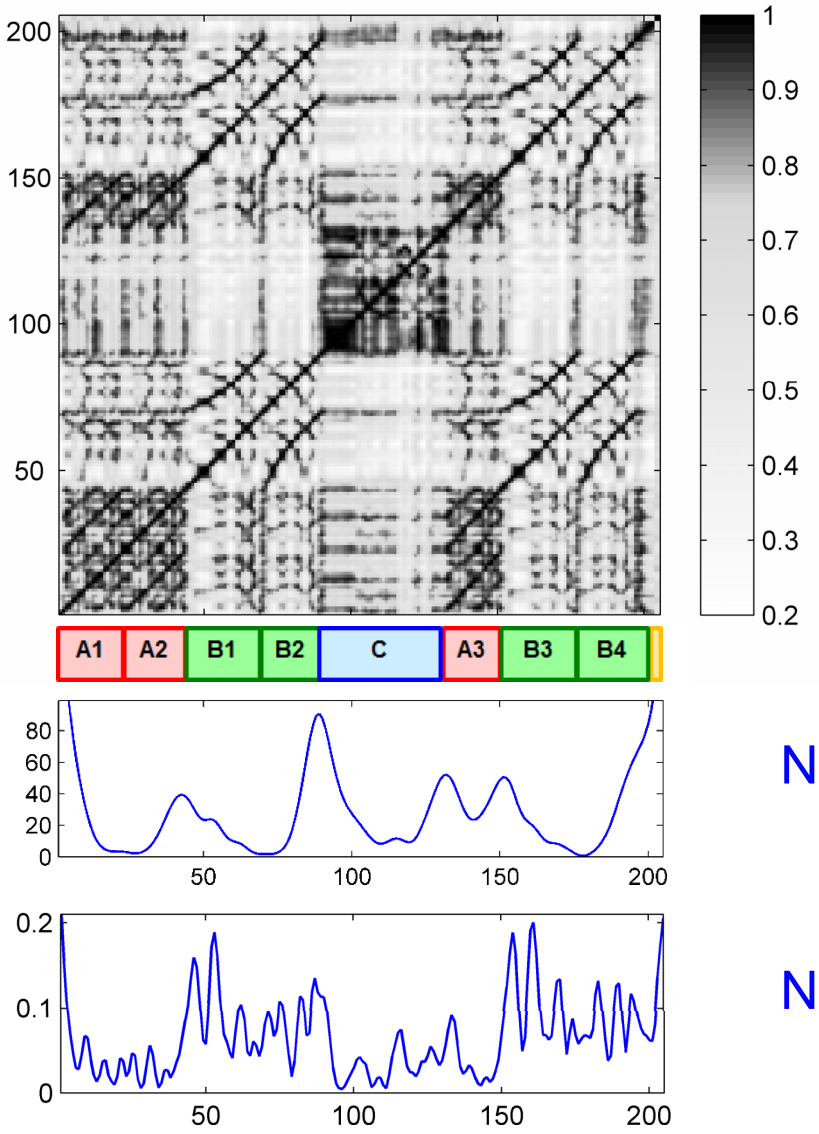


Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

Novelty function using

Novelty-Based Segmentation



Idea (Foote):

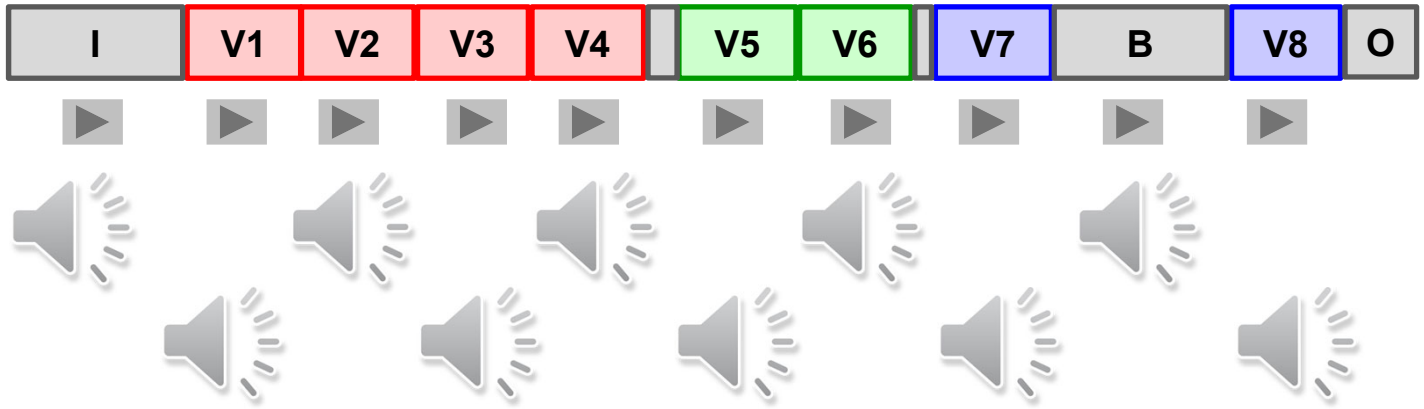
Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

Novelty function using 

Novelty function using 

SSM Enhancement

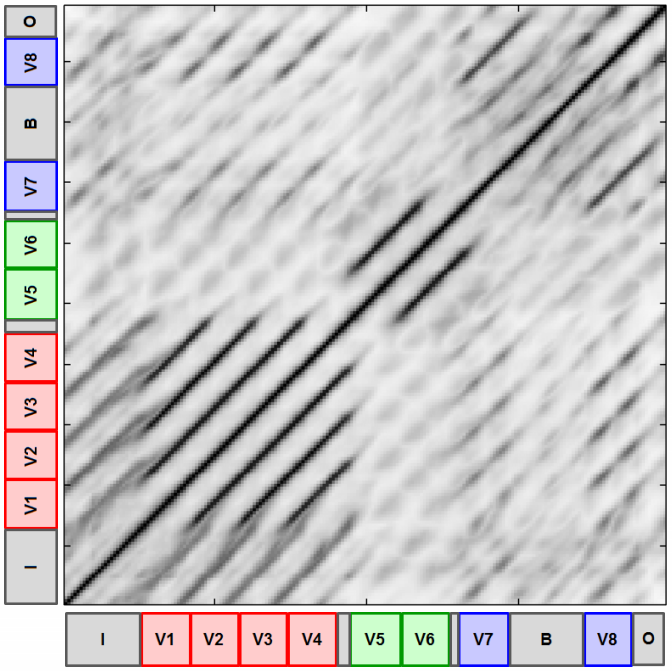
Example: Zager & Evans “In The Year 2525”



SSM Enhancement

Example: Zager & Evans “In The Year 2525”

Missing relations because of transposed sections

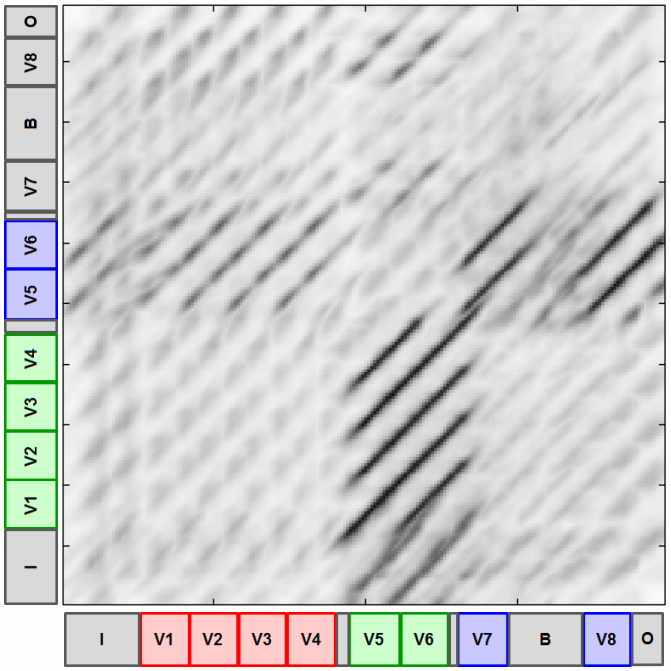


SSM Enhancement

Example: Zager & Evans “In The Year 2525”

Idea: Cyclic shift of one of the chroma sequences

One semitone up

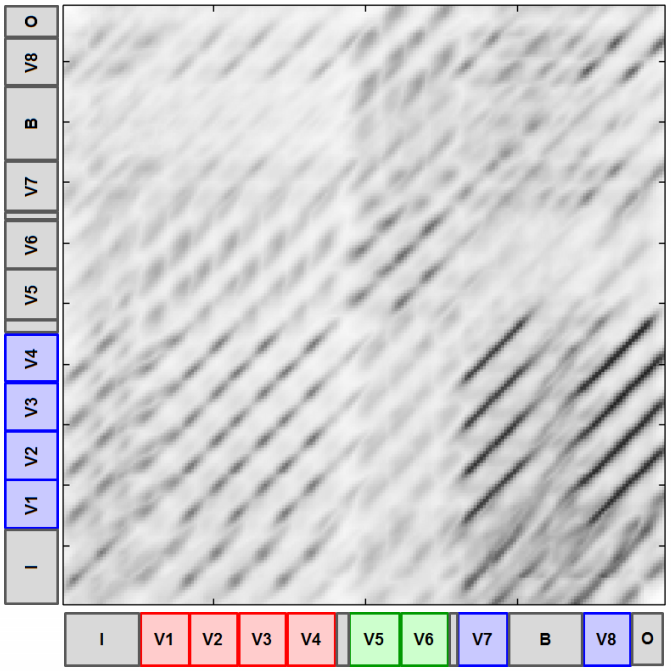


SSM Enhancement

Example: Zager & Evans “In The Year 2525”

Idea: Cyclic shift of one of the chroma sequences

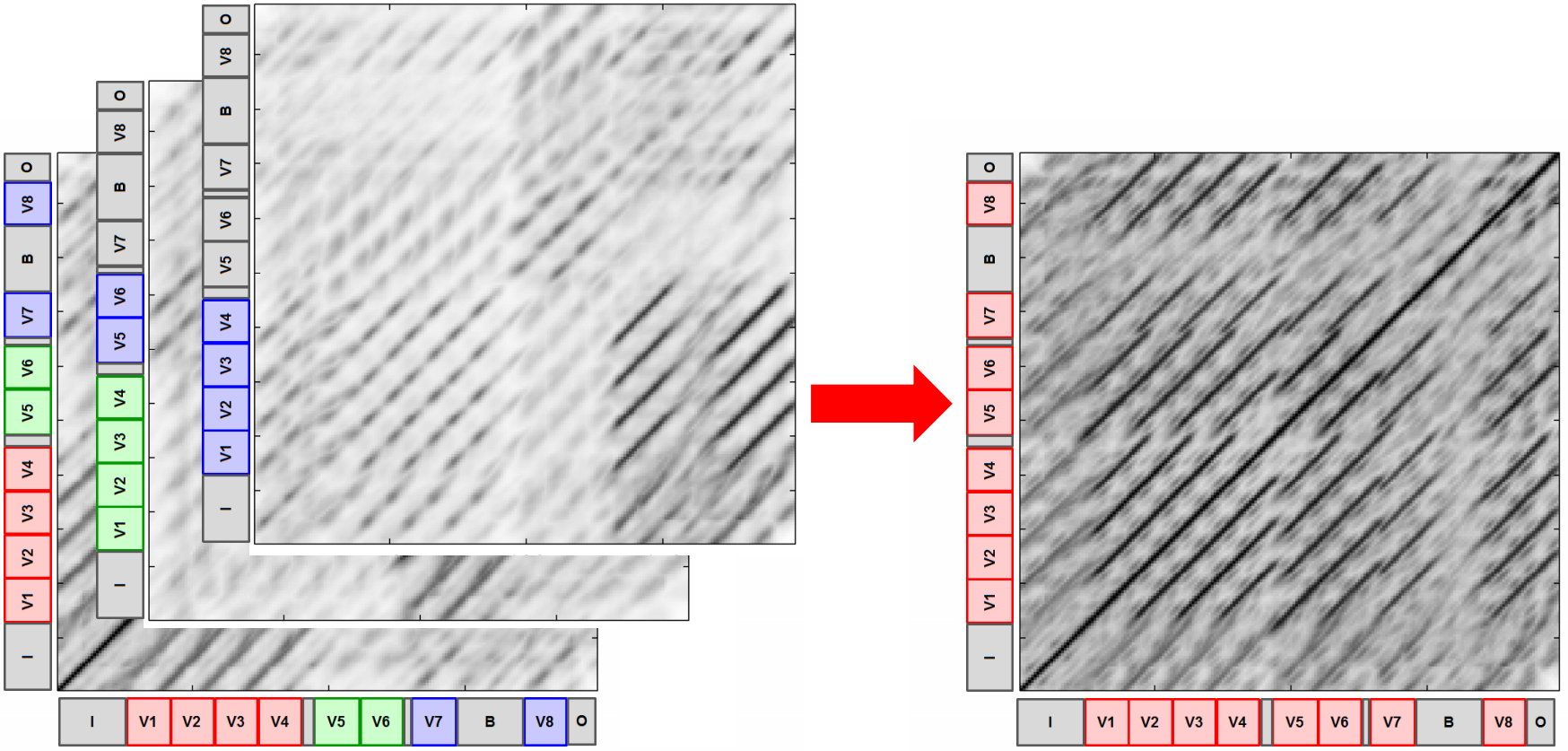
Two semitones up



SSM Enhancement

Example: Zager & Evans “In The Year 2525”

Idea: Overlay & Maximize → Transposition-invariant SSM

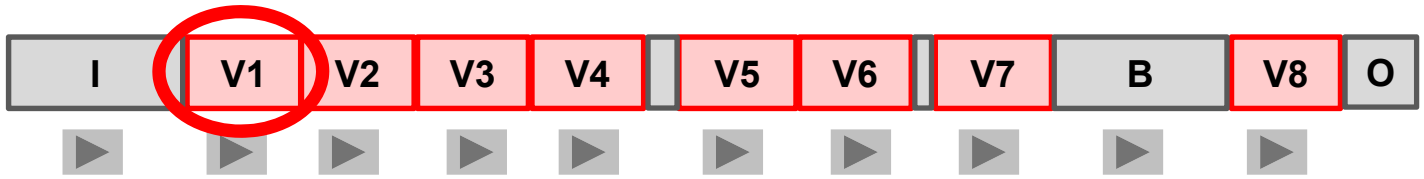


Audio Thumbnailing

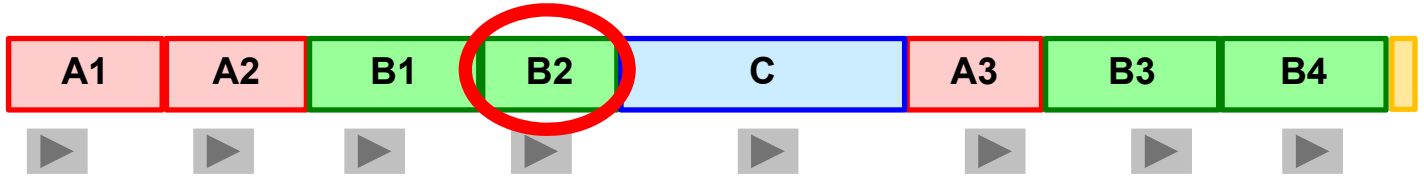
Audio Thumbnailing

General goal: Determine the most representative section ("Thumbnail") of a given music recording.

Example: Zager & Evans "In The Year 2525"



Example: Brahms Hungarian Dance No. 5 (Ormandy)



Thumbnail is often assumed to be the most repetitive segment

Audio Thumbnailing

Two steps

1. Path extraction

2. Grouping

Both steps are problematic!

- Paths of poor quality (fragmented, gaps)
- Block-like structures
- Curved paths

- Noisy relations
  (missing, distorted, overlapping)
- Transitivity computation difficult

Main idea: Do both, path extraction and grouping, jointly

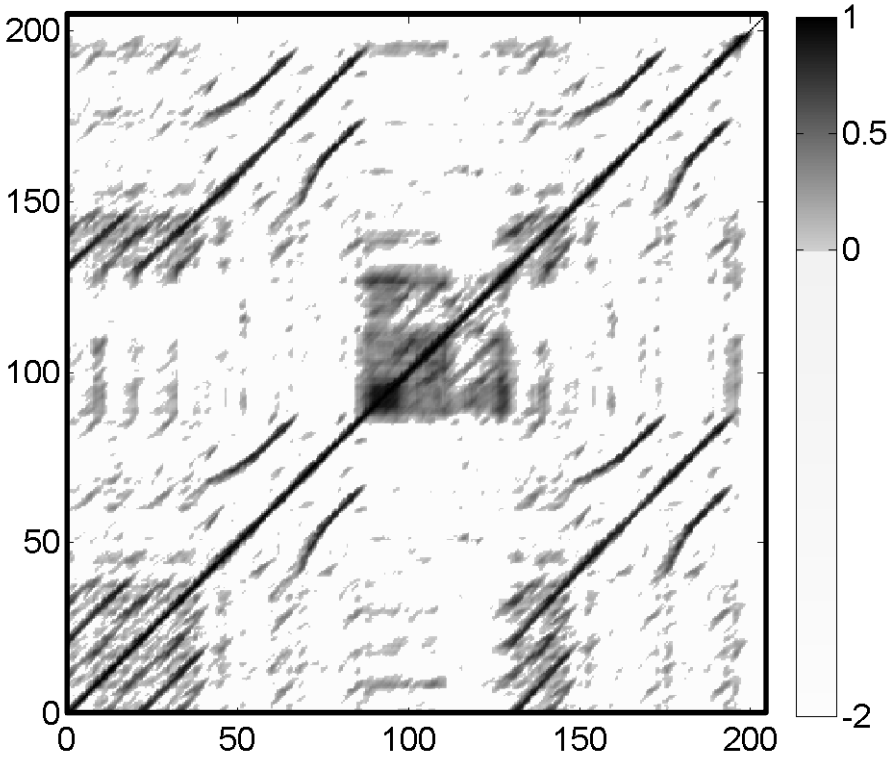
- One optimization scheme for both steps
- Stabilizing effect
- Efficient

Audio Thumbnailing

Main idea: Do both path extraction and grouping jointly

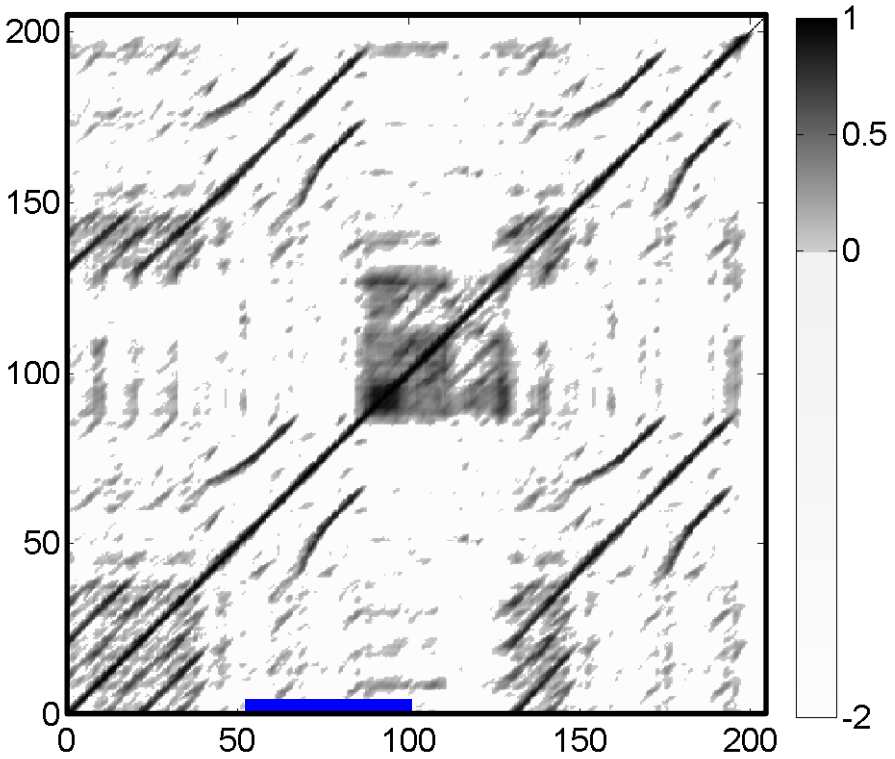
- For each audio segment we define a fitness value
- This fitness value expresses “how well” the segment explains the entire audio recording
- The segment with the highest fitness value is considered to be the thumbnail
- As main technical concept we introduce the notion of a path family

Fitness Measure



Enhanced SSM

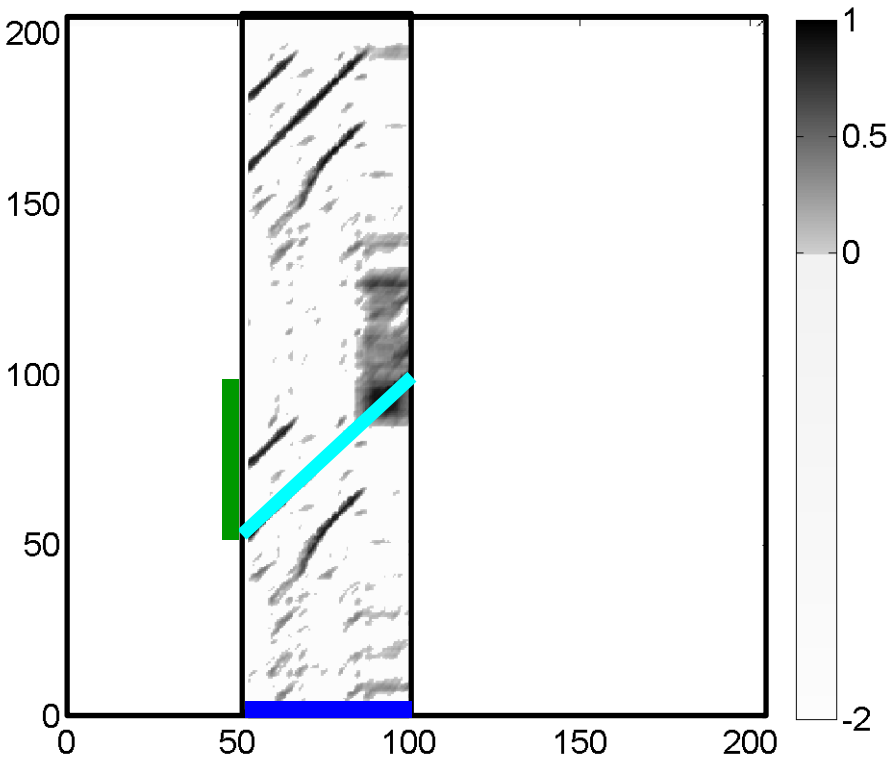
Fitness Measure



Path over segment

- Consider a fixed segment

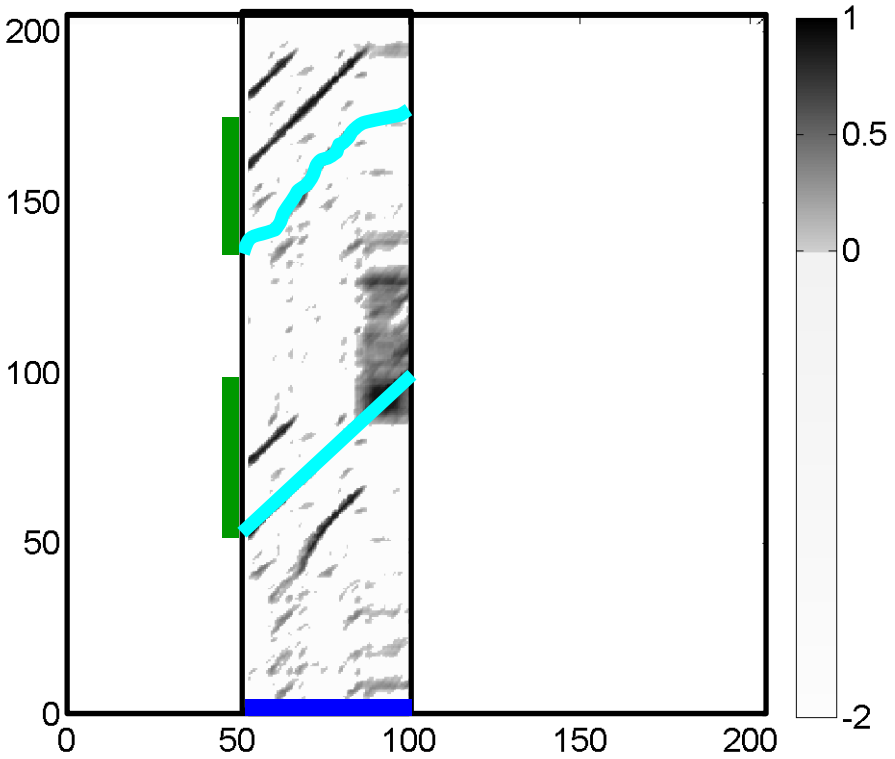
Fitness Measure



Path over segment

- Consider a fixed segment
- Path over segment
- Induced segment
- Score is high

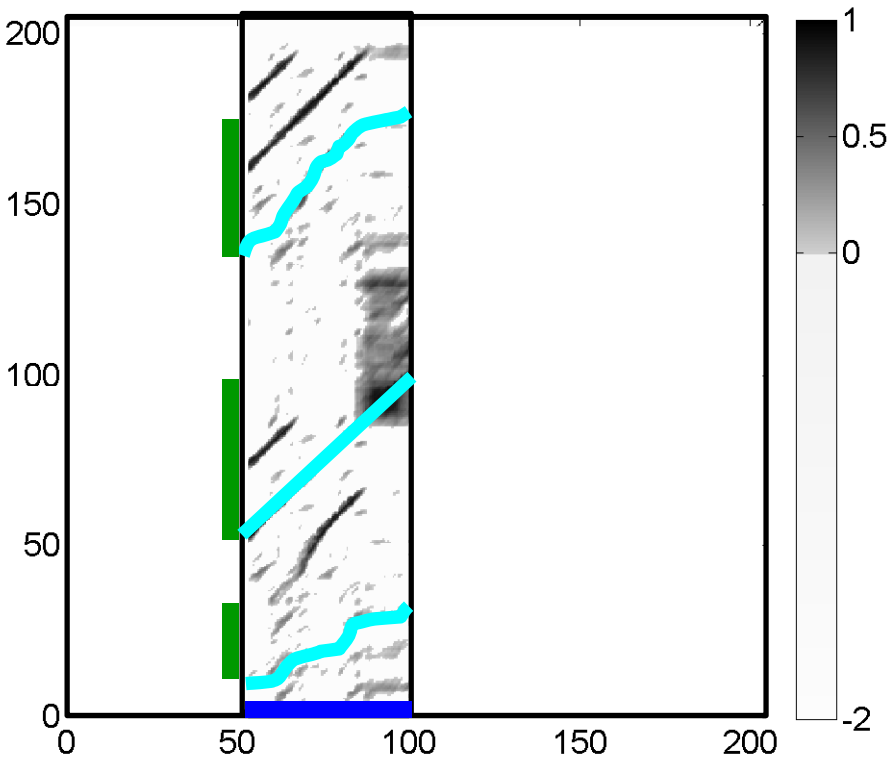
Fitness Measure



Path over segment

- Consider a fixed segment
- Path over segment
- Induced segment
- Score is high
- A second path over segment
- Induced segment
- Score is not so high

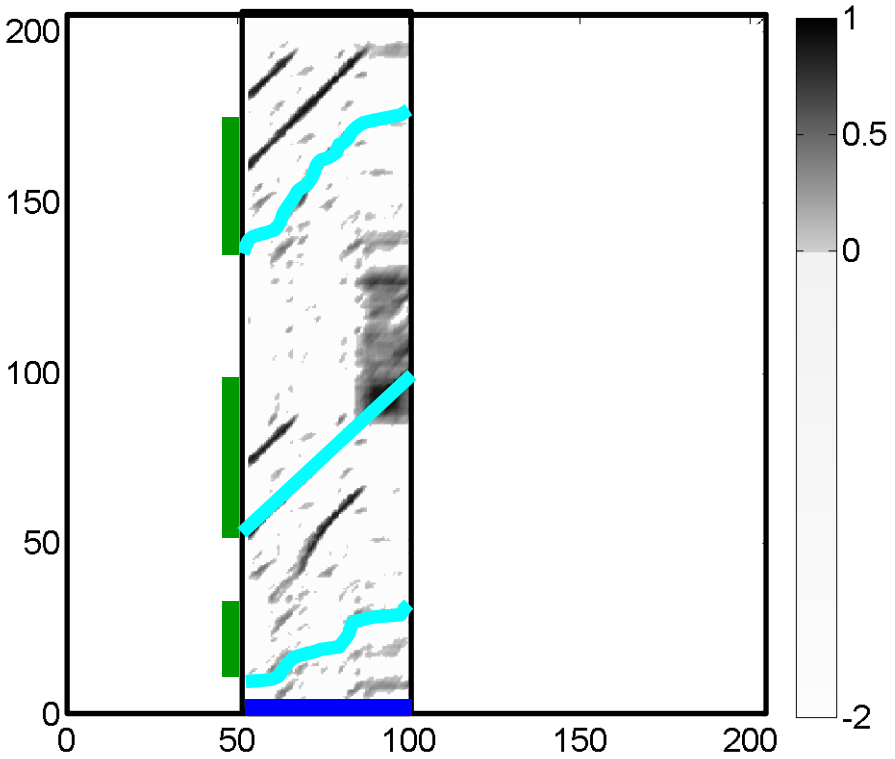
Fitness Measure



Path over segment

- Consider a fixed segment
- Path over segment
- Induced segment
- Score is high
- A second path over segment
- Induced segment
- Score is not so high
- A third path over segment
- Induced segment
- Score is very low

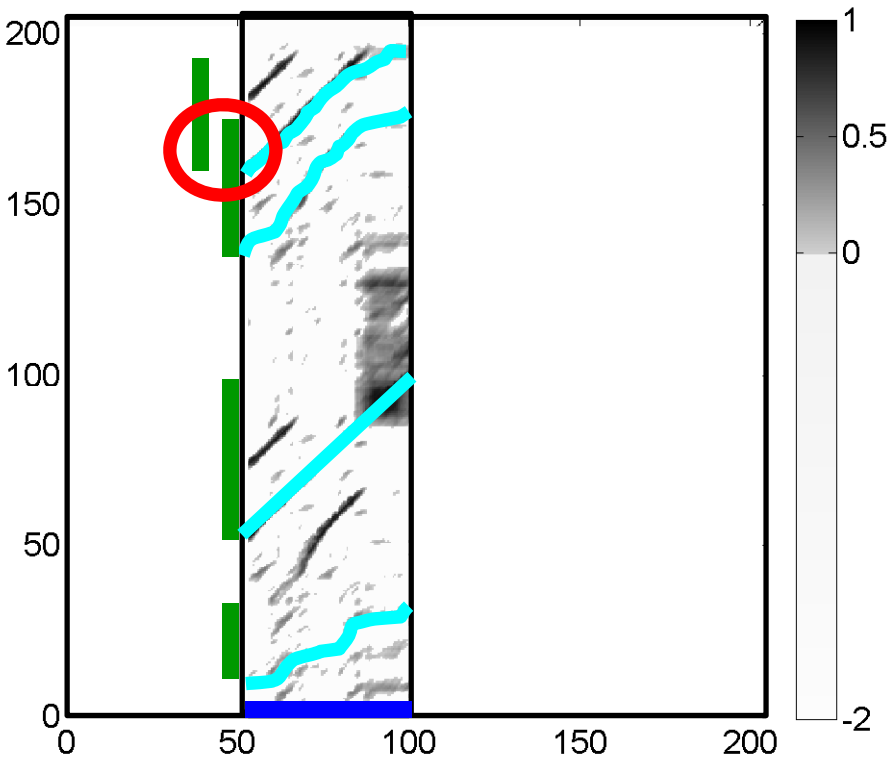
Fitness Measure



Path family

- Consider a fixed segment
- A path family over a segment is a family of paths such that the induced segments do not overlap.

Fitness Measure

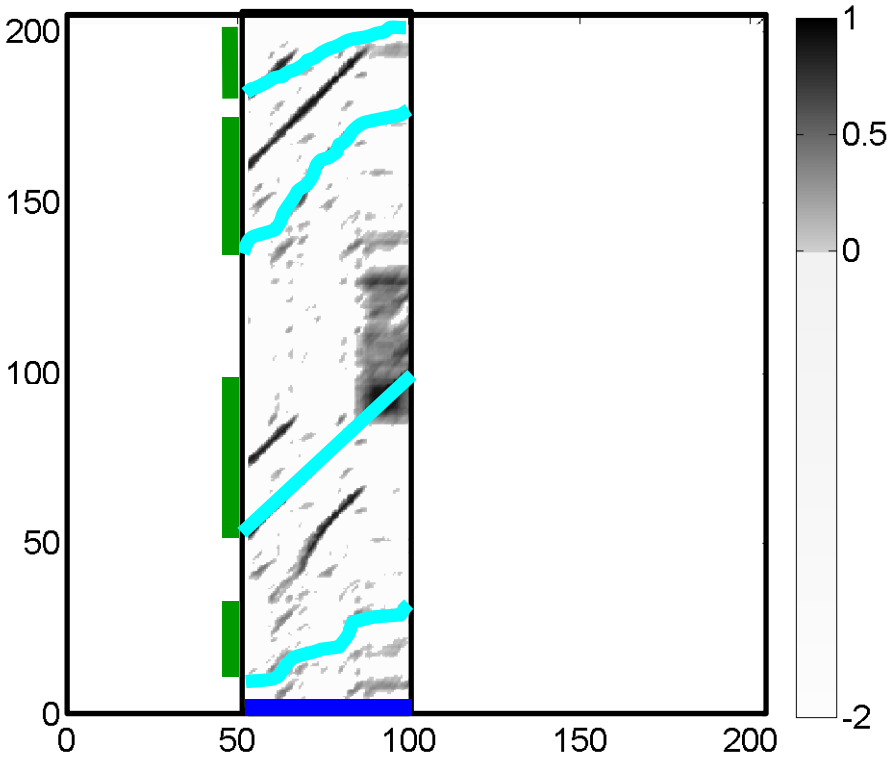


Path family

- Consider a fixed segment
- A path family over a segment is a family of paths such that the induced segments do not overlap.

This is not a path family!

Fitness Measure



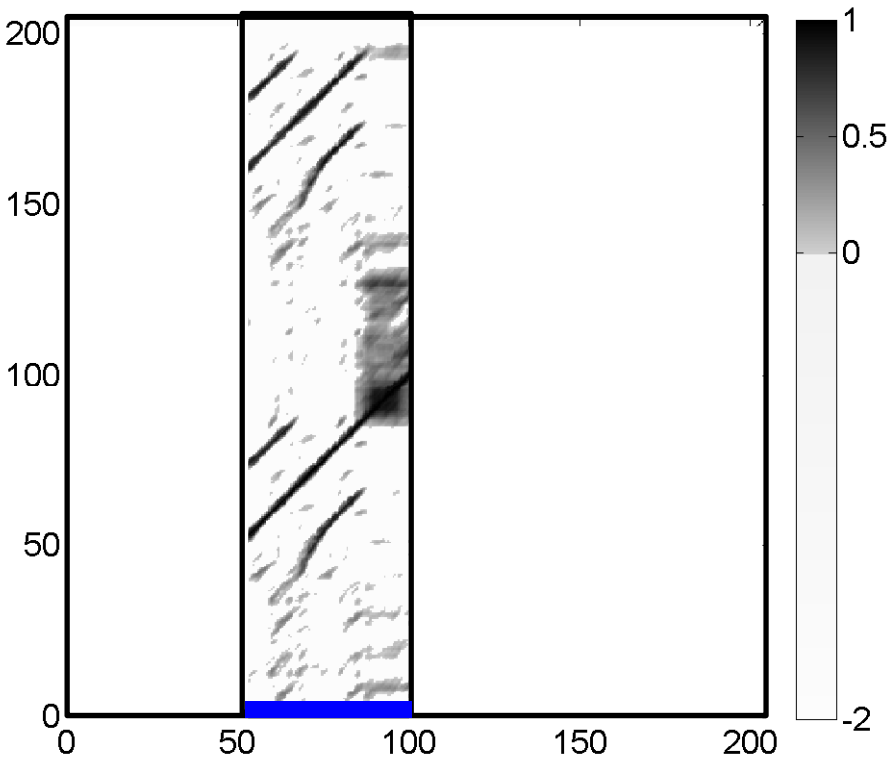
Path family

- Consider a fixed segment
- A path family over a segment is a family of paths such that the induced segments do not overlap.

This is a path family!

(Even though not a good one)

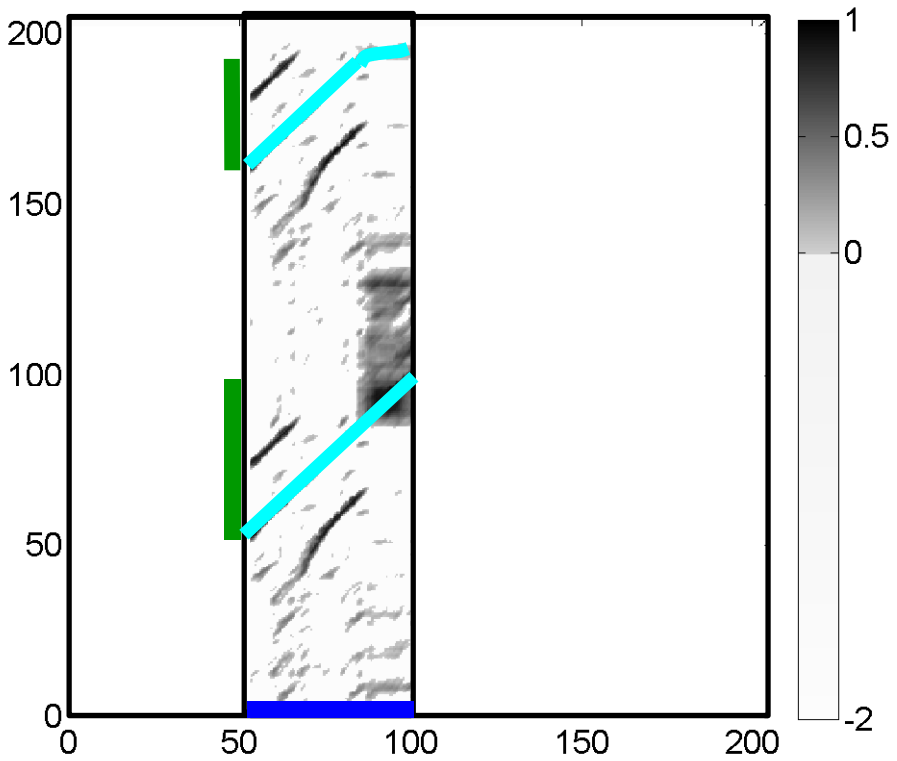
Fitness Measure



Optimal path family

- Consider a fixed segment

Fitness Measure



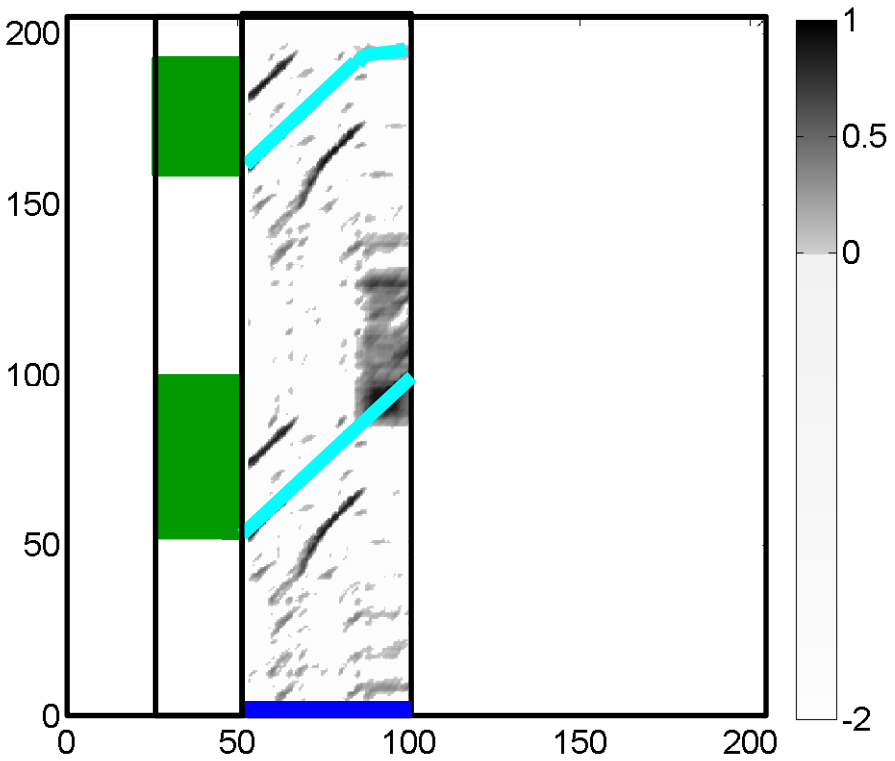
Optimal path family

- Consider a fixed segment
- Consider over the segment the optimal path family, i.e., the path family having maximal overall score.
- Call this value:

Score(segment)

Note: This optimal path family can be computed using dynamic programming.

Fitness Measure



Optimal path family

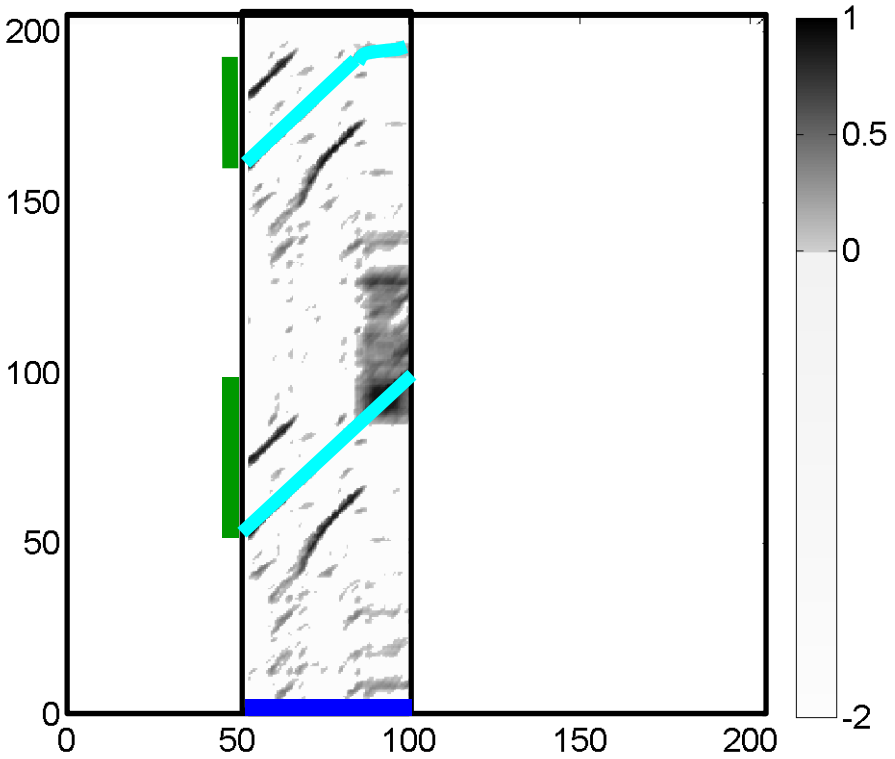
- Consider a fixed segment
- Consider over the segment the optimal path family, i.e., the path family having maximal overall score.
- Call this value:

  Score(segment)

- Furthermore consider the amount covered by the induced segments.
- Call this value:

  Coverage(segment)

Fitness Measure



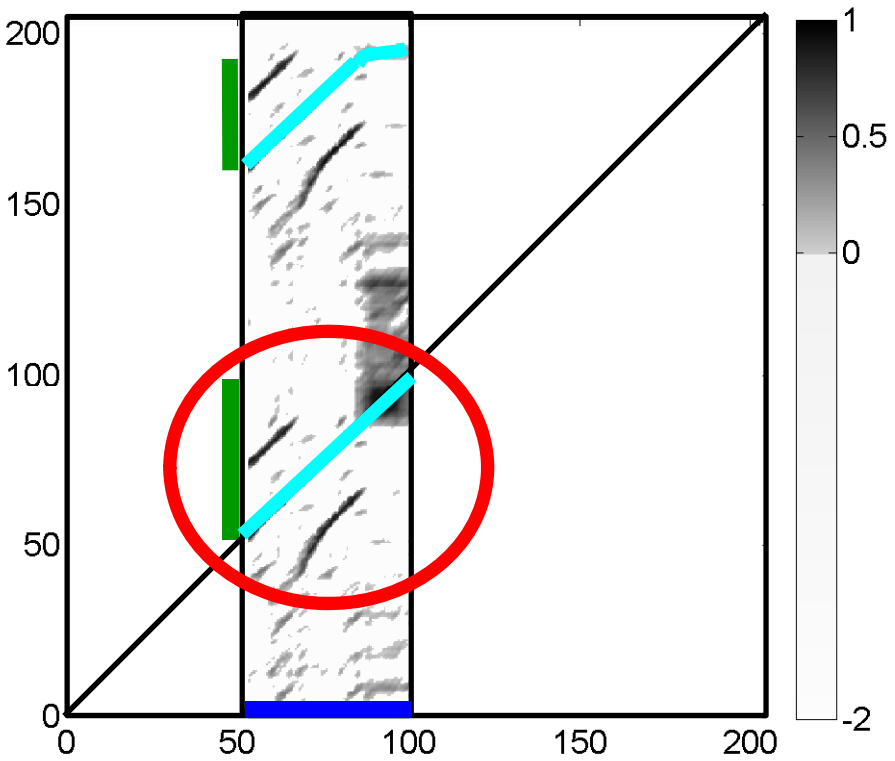
P := Score(segment)

R := Coverage(segment)

Fitness

- Consider a fixed segment

Fitness Measure



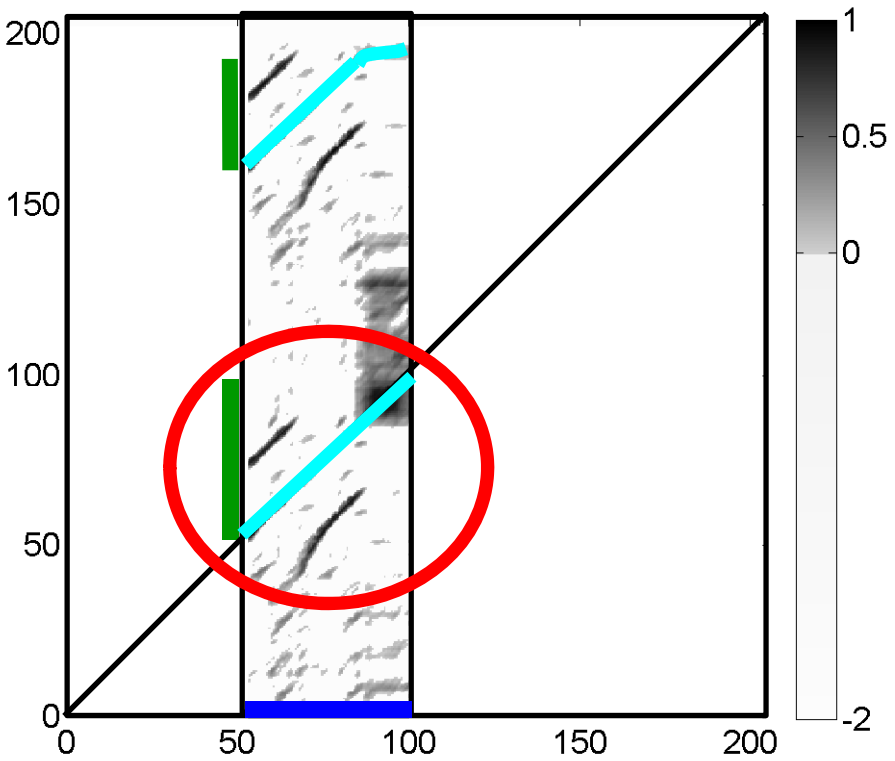
P := Score(segment)

R := Coverage(segment)

Fitness

- Consider a fixed segment
- Self-explanation are trivial!

Fitness Measure



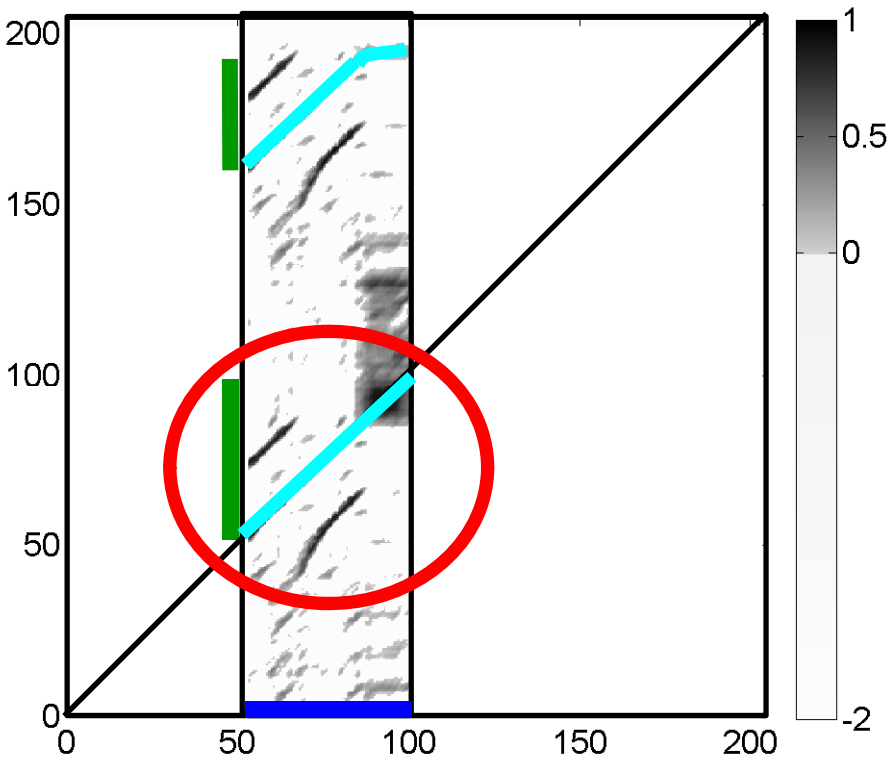
Fitness

- Consider a fixed segment
- Self-explanation are trivial!
- Subtract length of segment

P := Score(segment) - length(segment)

R := Coverage(segment) - length(segment)

Fitness Measure



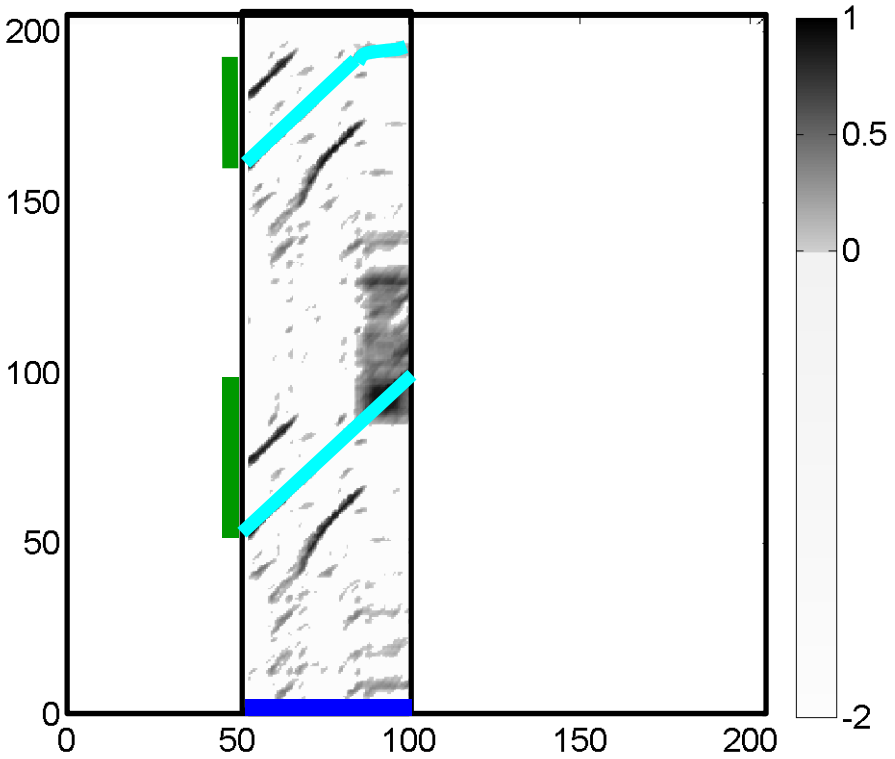
Fitness

- Consider a fixed segment
- Self-explanation are trivial!
- Subtract length of segment
- Normalization

P := Normalize(Score(segment) - length(segment)) $\in [0,1]$

R := Normalize(Coverage(segment) - length(segment)) $\in [0,1]$

Fitness Measure



Fitness

- Consider a fixed segment

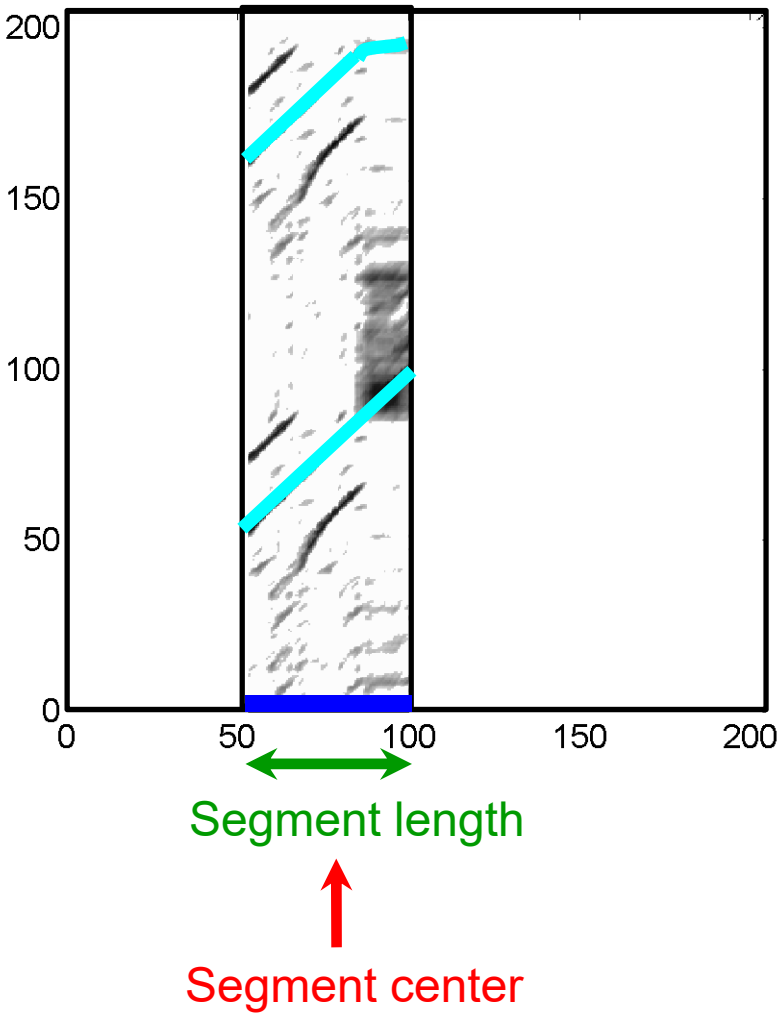
Fitness(segment)

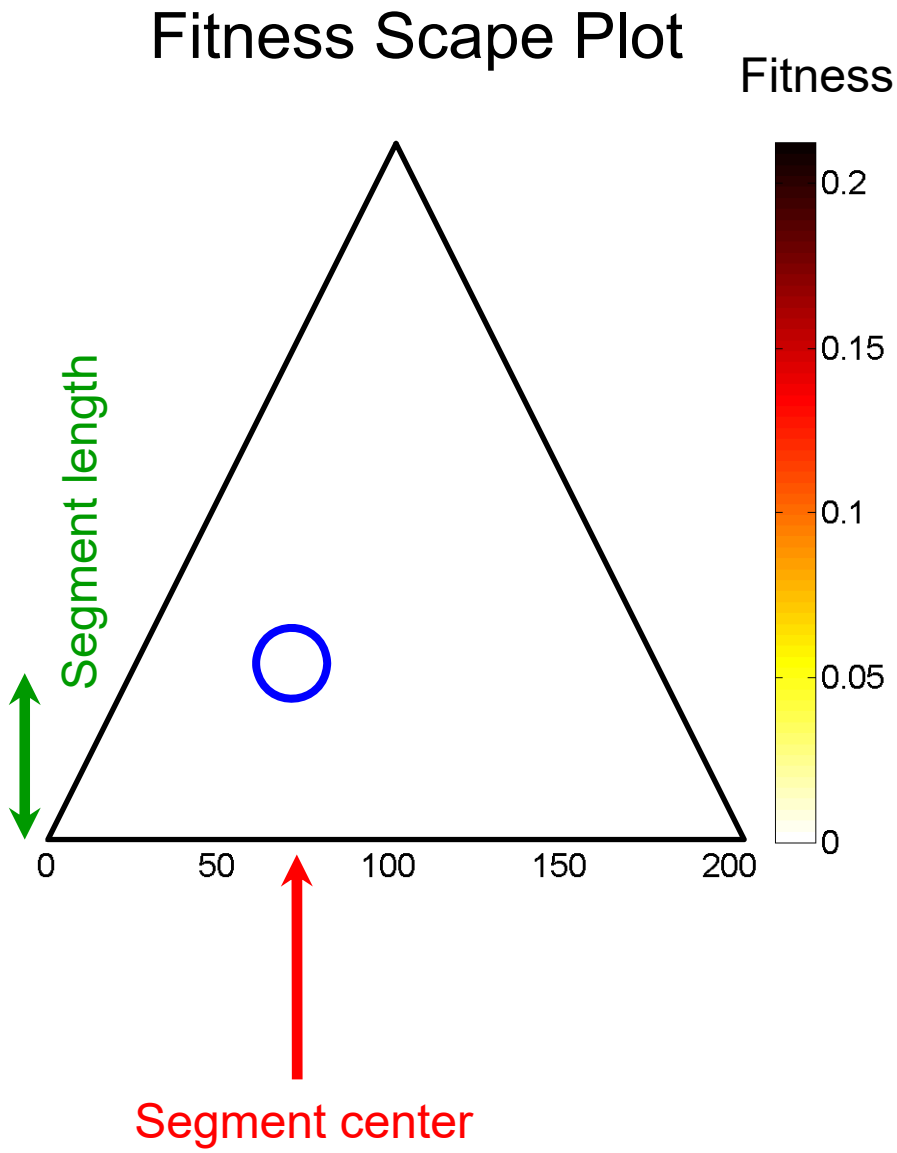
F := 2 • P • R / (P + R)

P := Normalize(Score(segment) - length(segment)) $\in [0,1]$

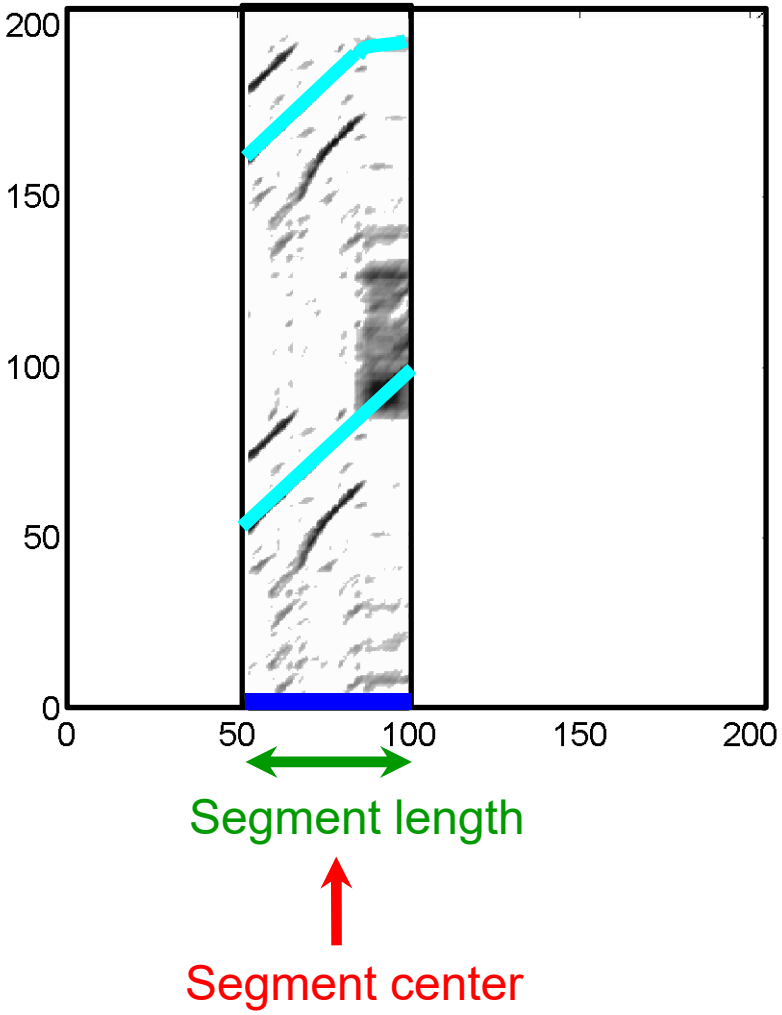
R := Normalize(Coverage(segment) - length(segment)) $\in [0,1]$

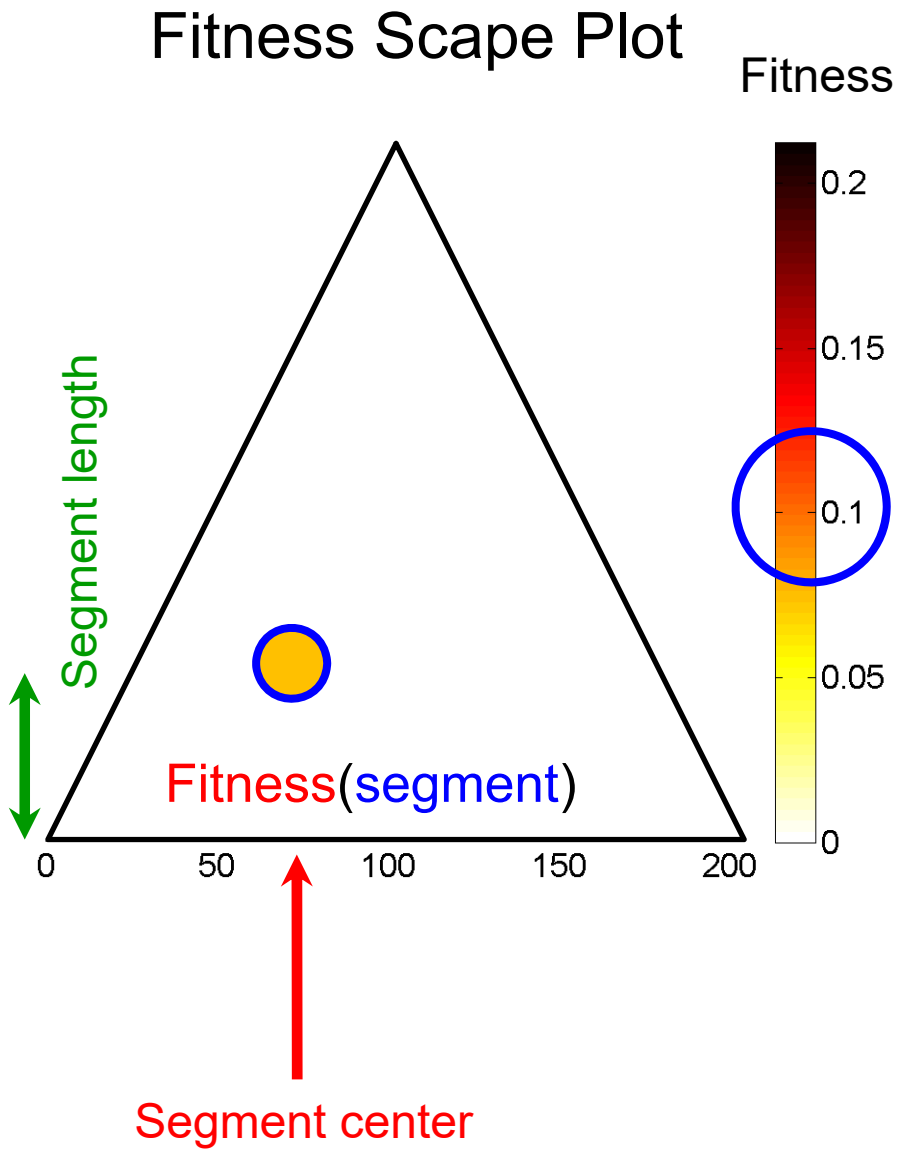
Thumbnail


200
150
100
50
0
0
50
100
150
200
Segment length
Segment center

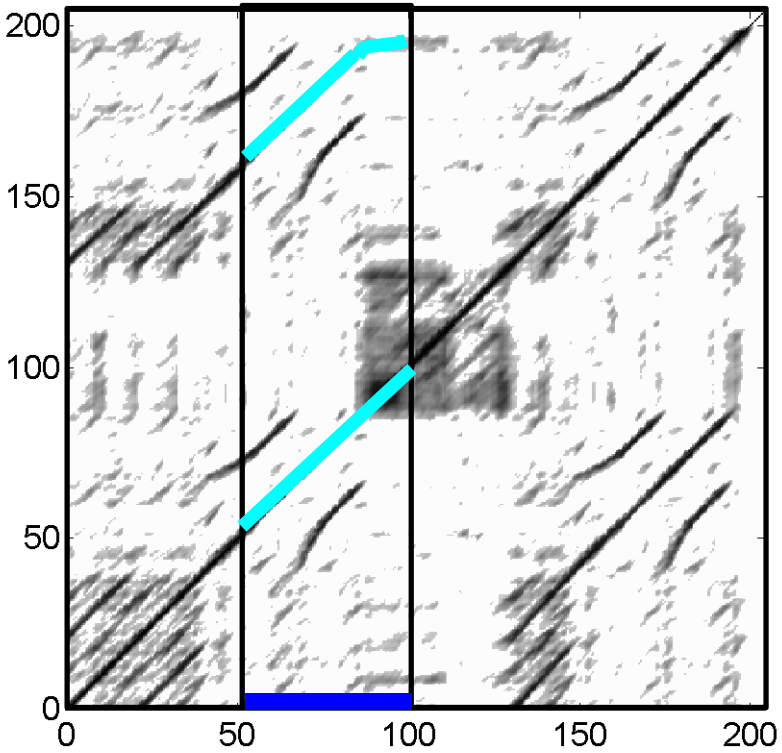

Fitness Scape Plot
Fitness
0.2
0.15
0.1
0.05
0
Segment length
0
50
100
150
200
Segment center

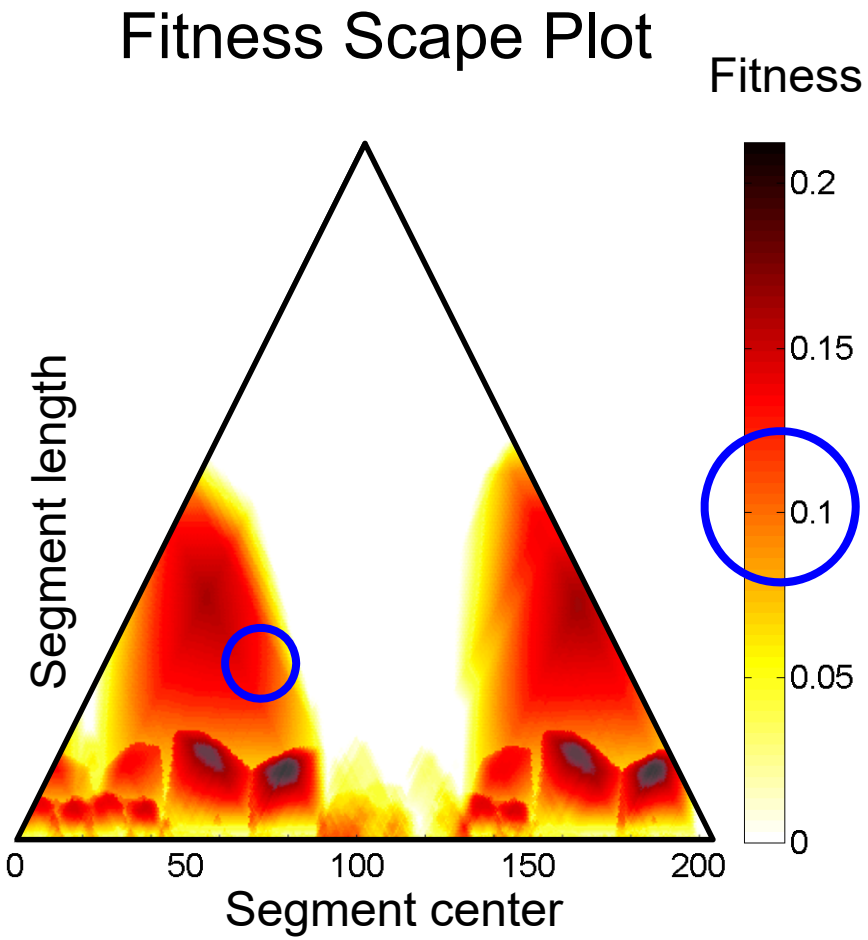
Thumbnail


200
150
100
50
0
0
50
100
150
200
Segment length
Segment center

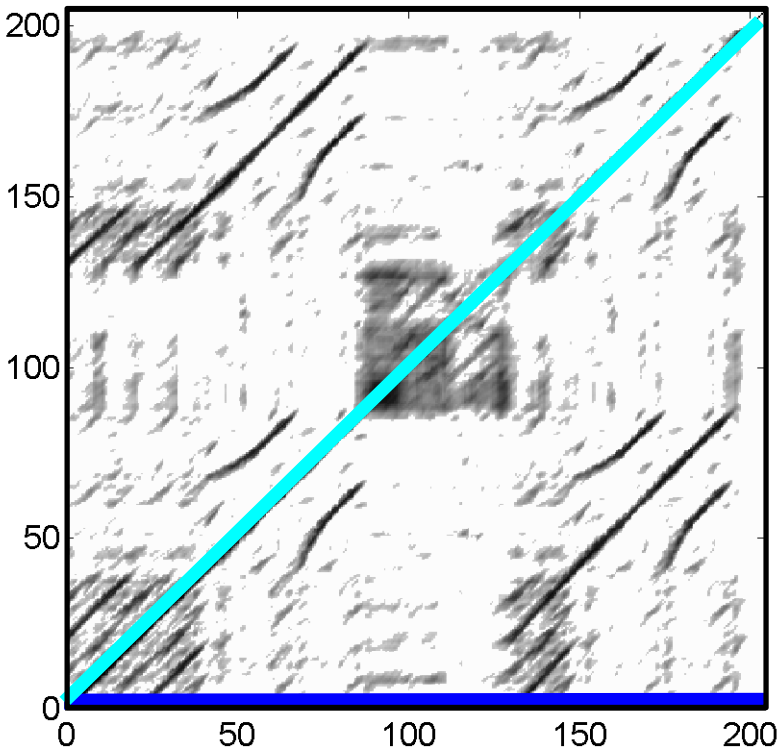

Fitness Scape Plot
Fitness
0.2
0.15
0.1
0.05
0
Segment length
Fitness(segment)
0
50
100
150
200
Segment center

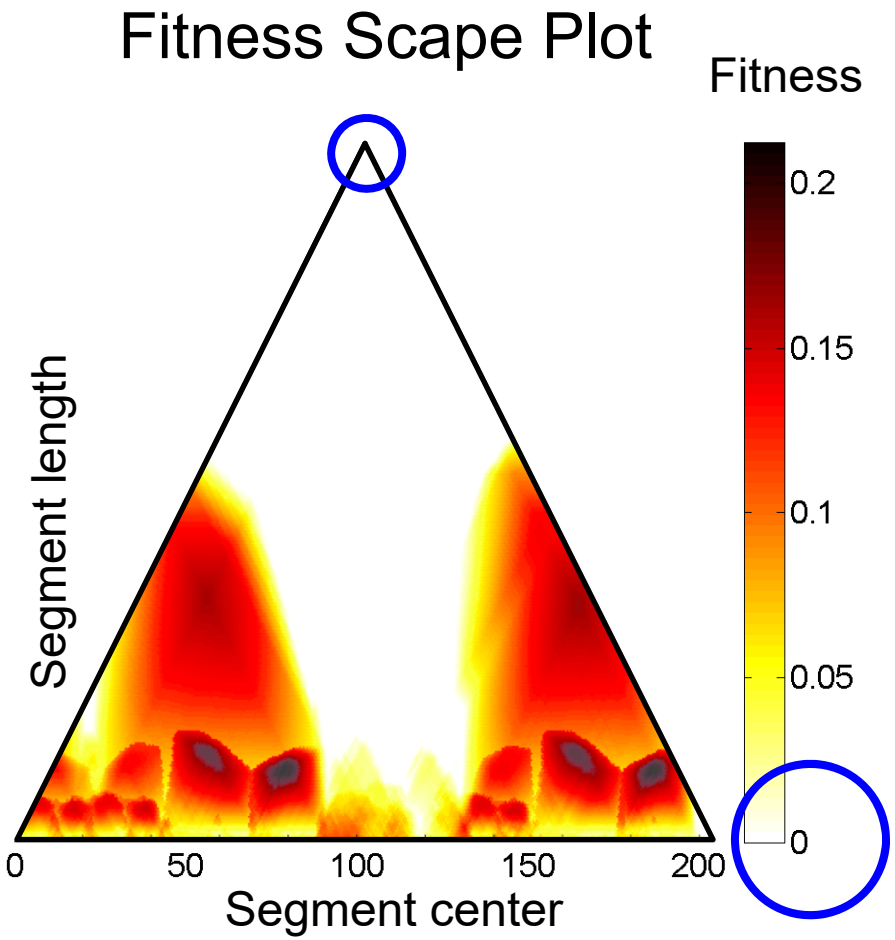
Thumbnail


200
150
100
50
0
0
50
100
150
200


Fitness Scape Plot
Fitness
0.2
0.15
0.1
0.05
0
Segment length
0
50
100
150
200
Segment center

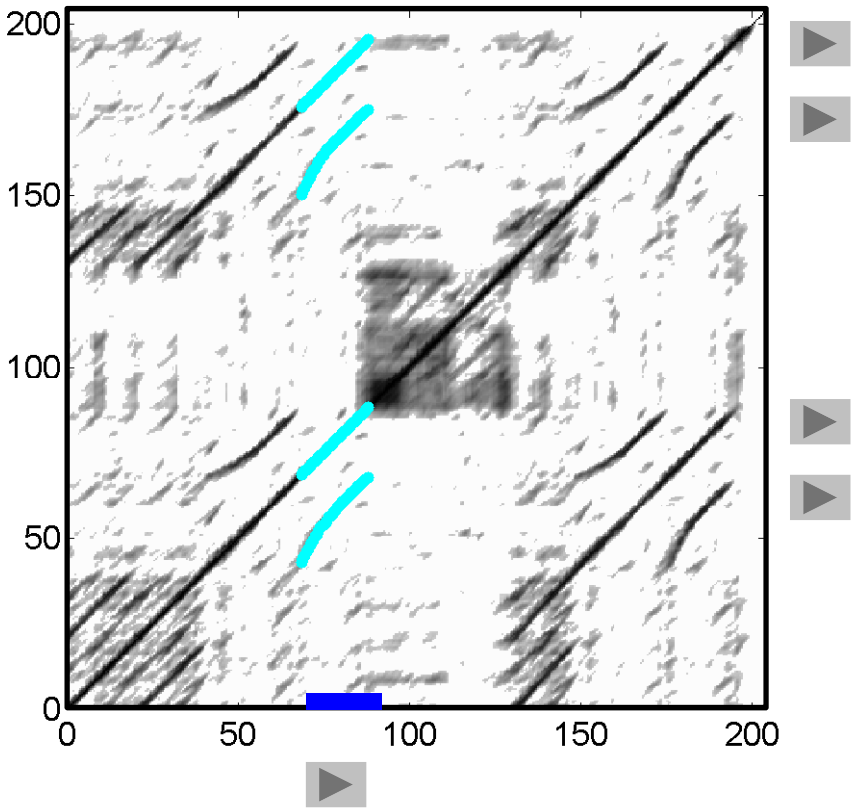
Thumbnail

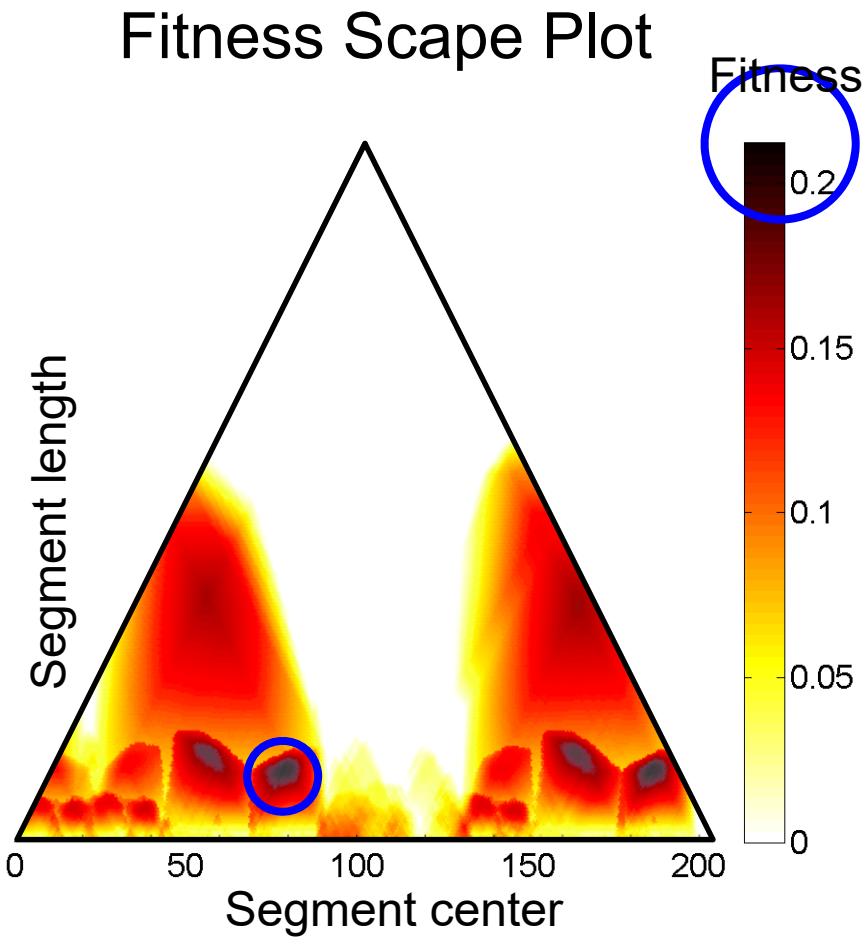




Note: Self-explanations are ignored → fitness is zero

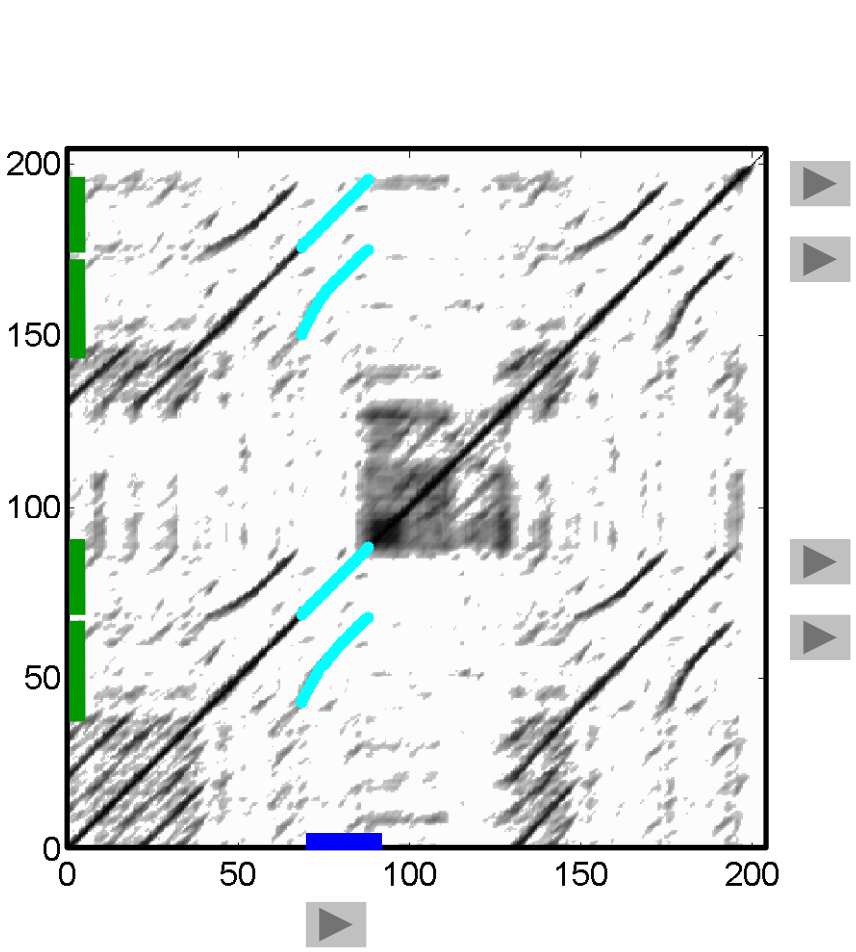
Thumbnail

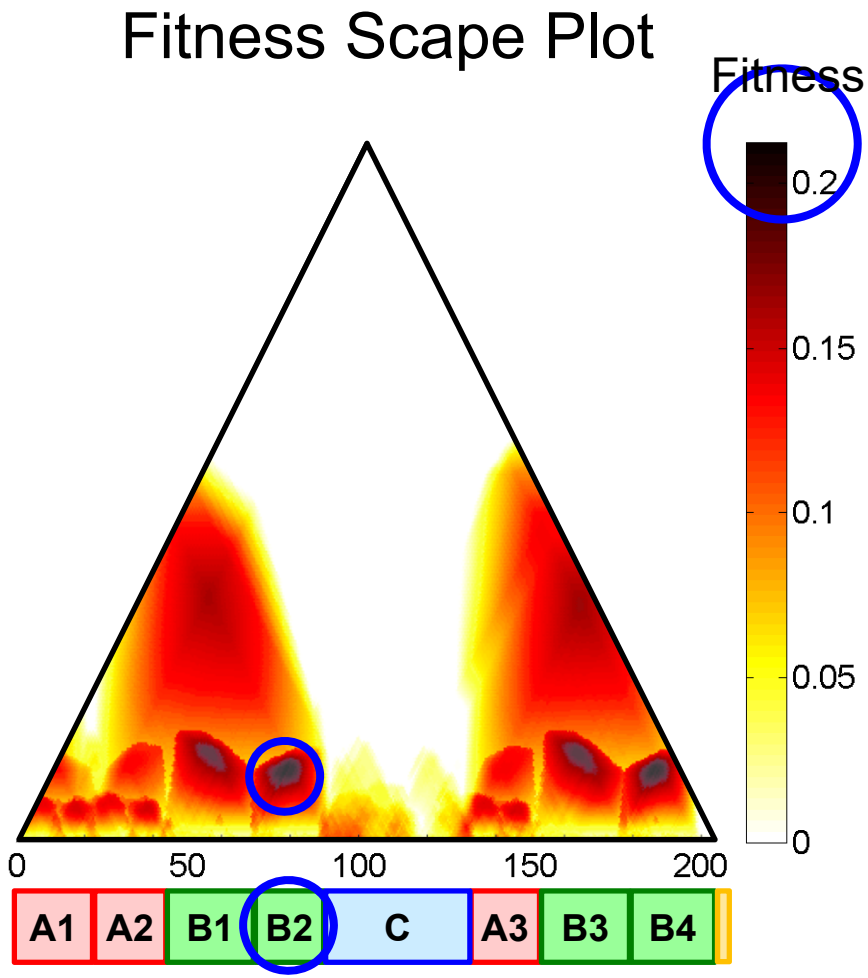




Thumbnail := segment having the highest fitness

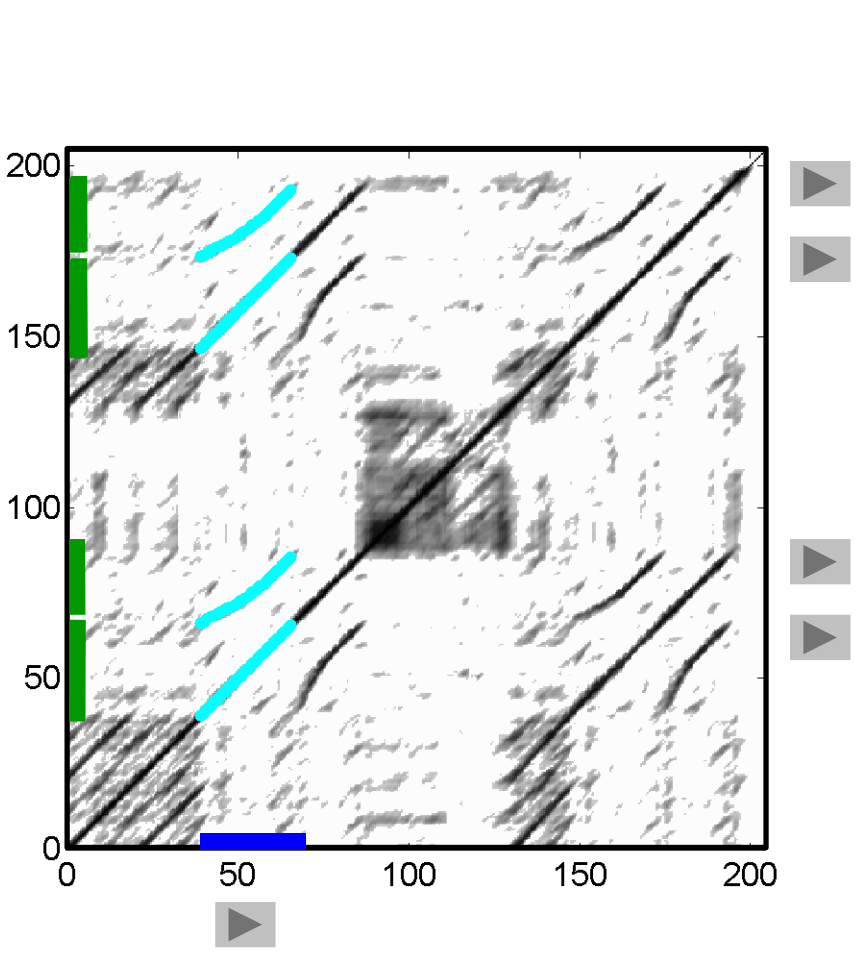
Thumbnail

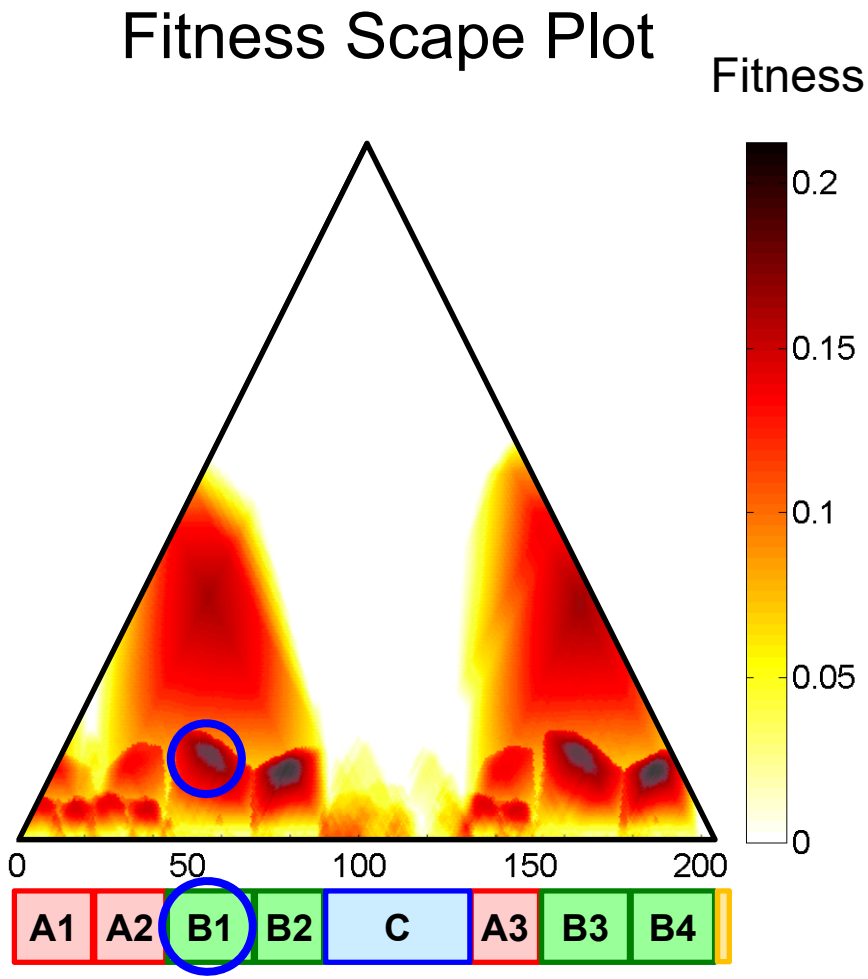




Example: Brahms Hungarian Dance No. 5 (Ormandy)

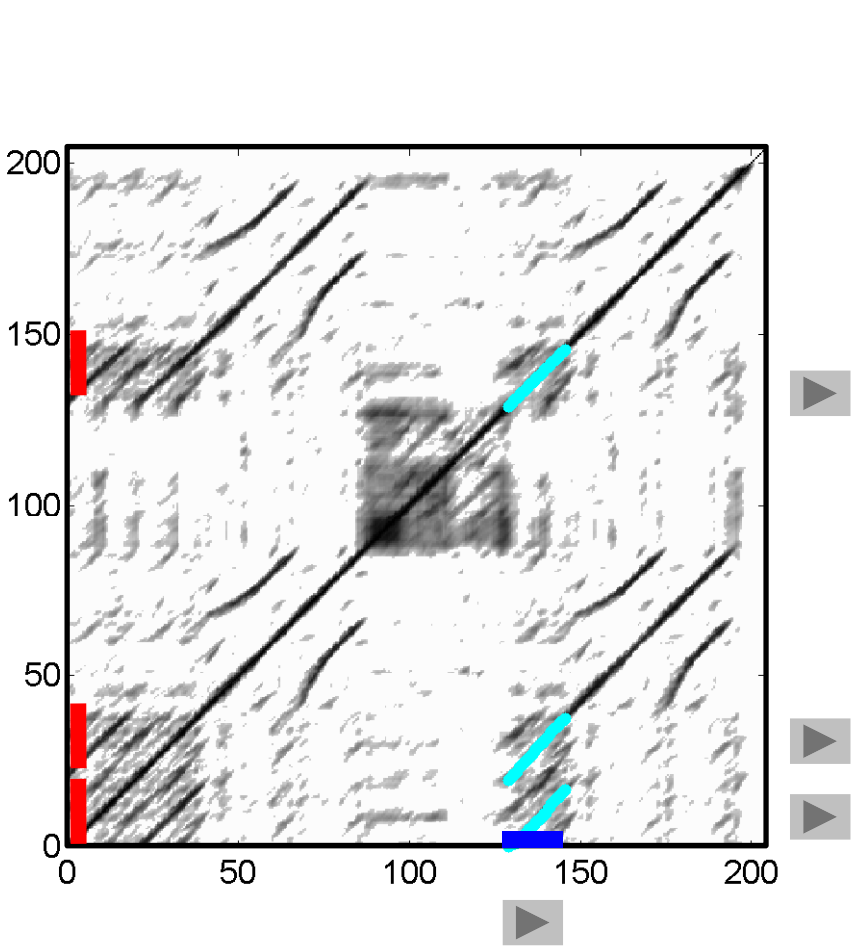
Thumbnail

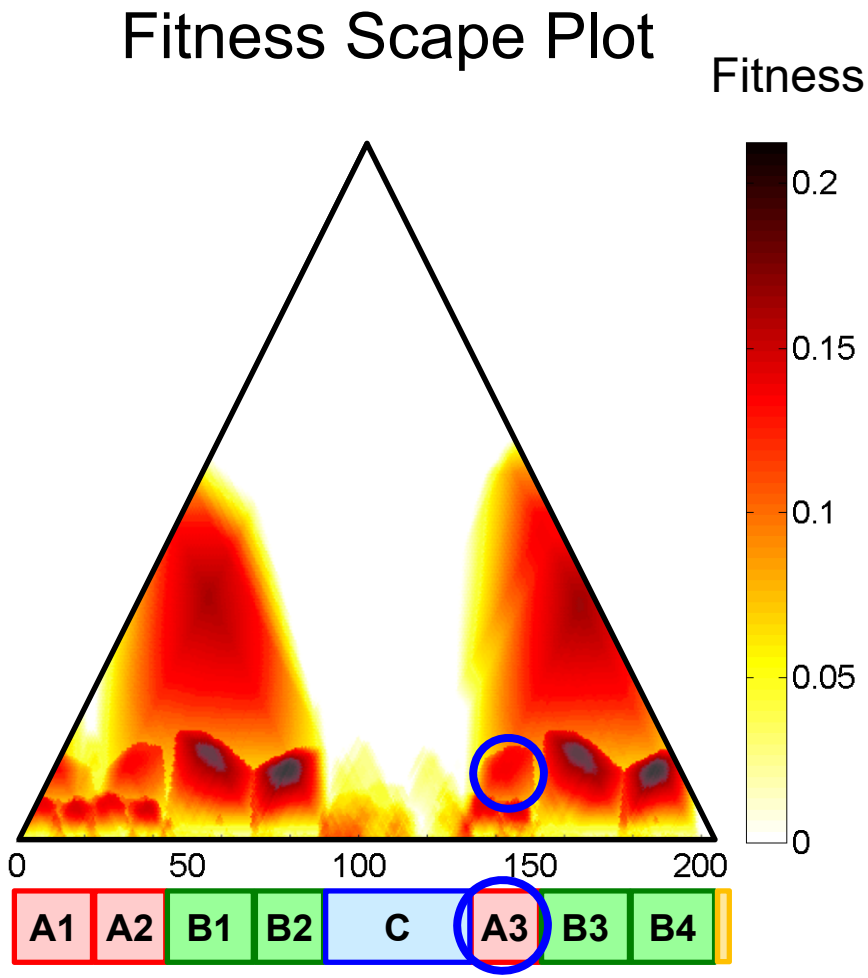




Example: Brahms Hungarian Dance No. 5 (Ormandy)

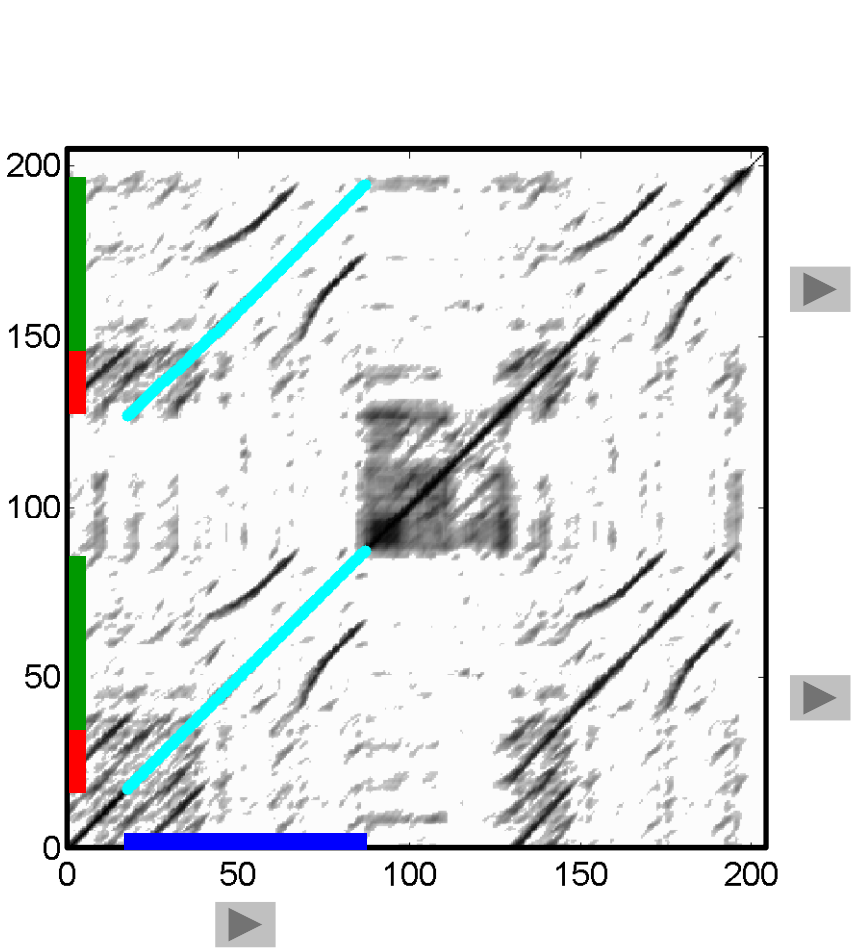
Thumbnail

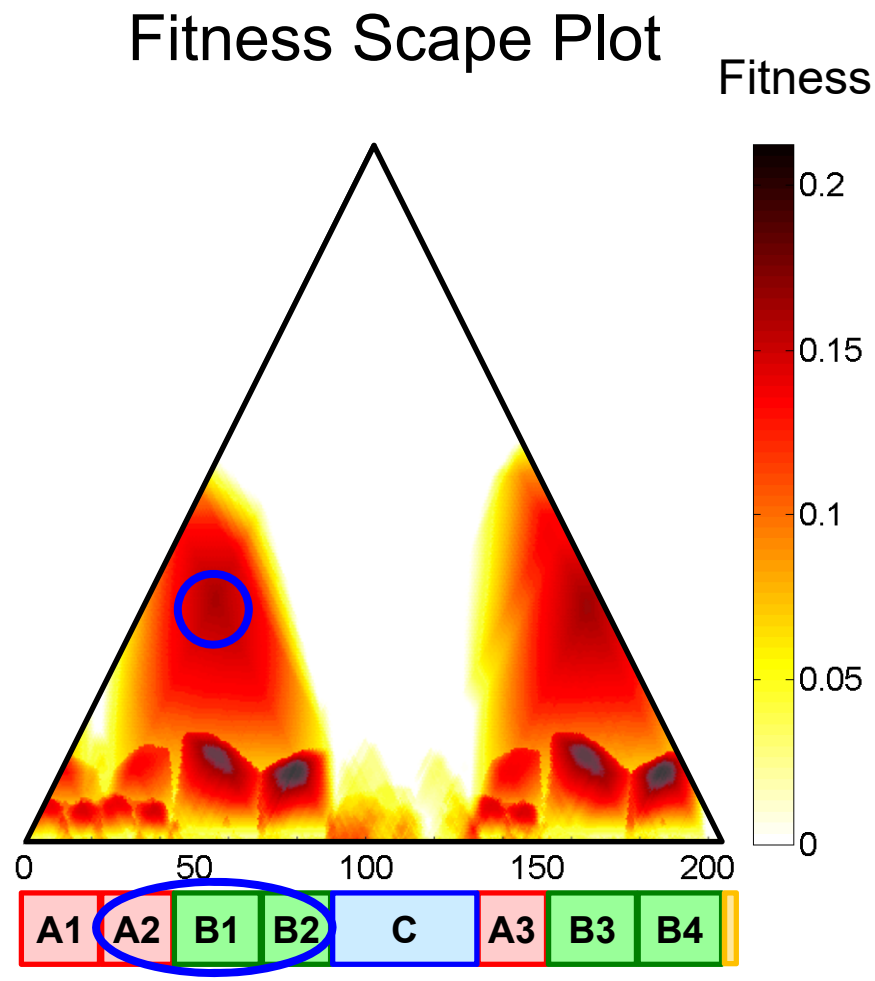




Example: Brahms Hungarian Dance No. 5 (Ormandy)

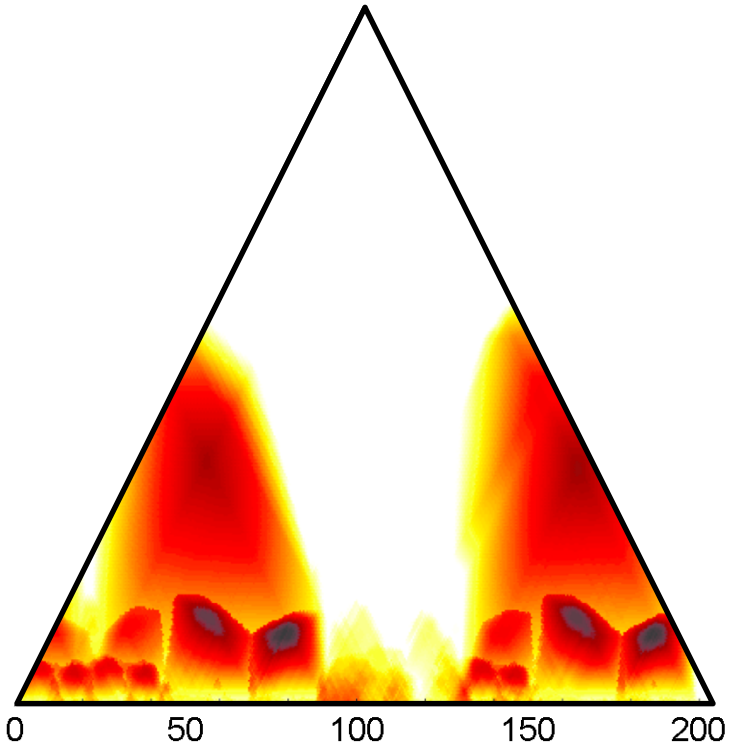
Thumbnail





Example: Brahms Hungarian Dance No. 5 (Ormandy)

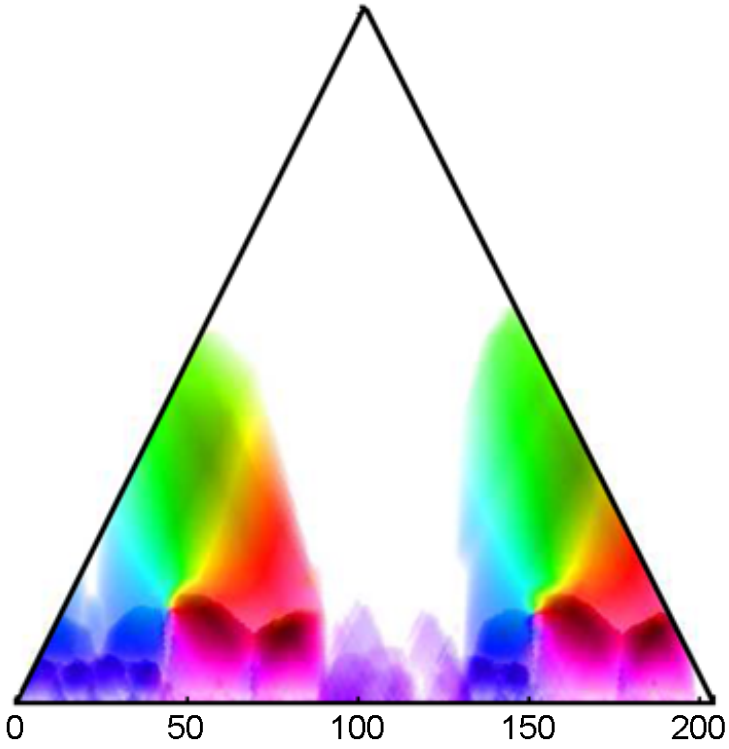
Scape Plot



Example: Brahms Hungarian Dance No. 5 (Ormandy)

Scape Plot

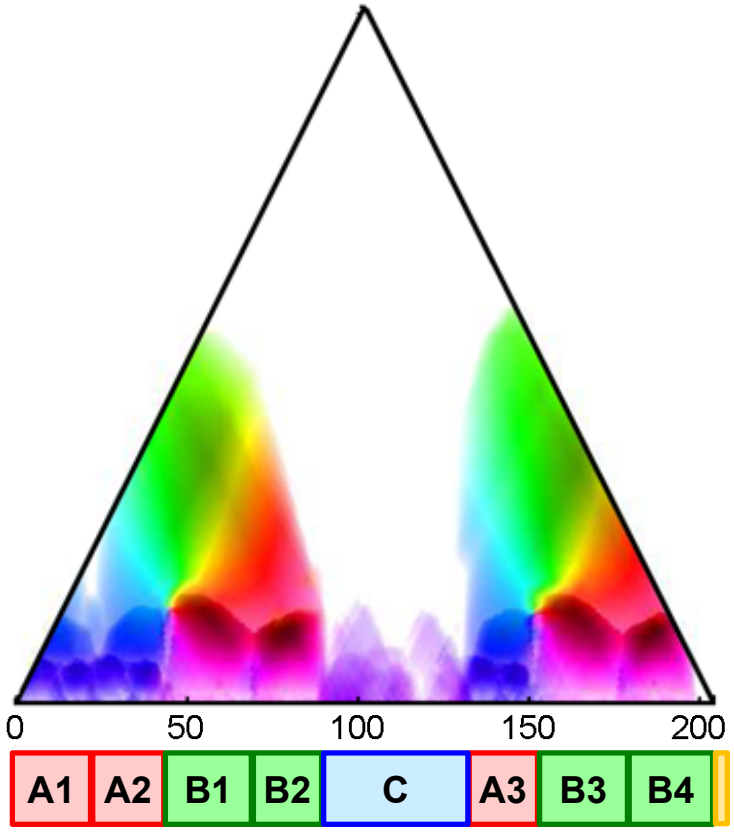
Coloring according to clustering result (grouping)



Example: Brahms Hungarian Dance No. 5 (Ormandy)

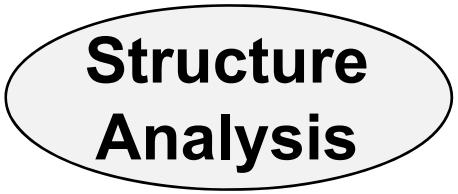
Scape Plot

Coloring according to clustering result (grouping)

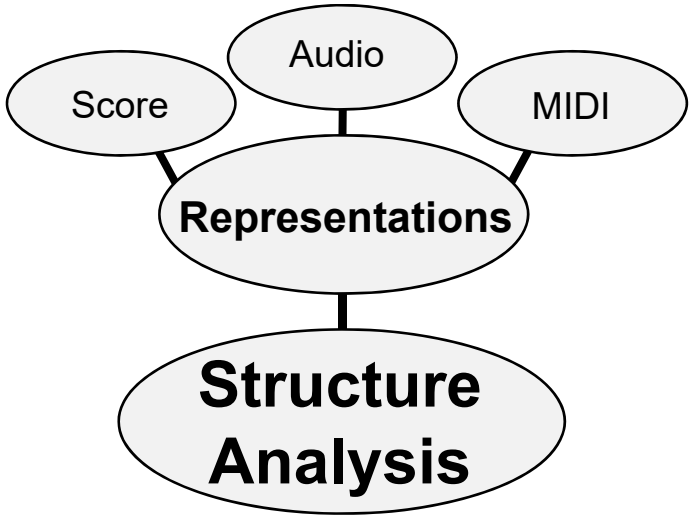


Example: Brahms Hungarian Dance No. 5 (Ormandy)

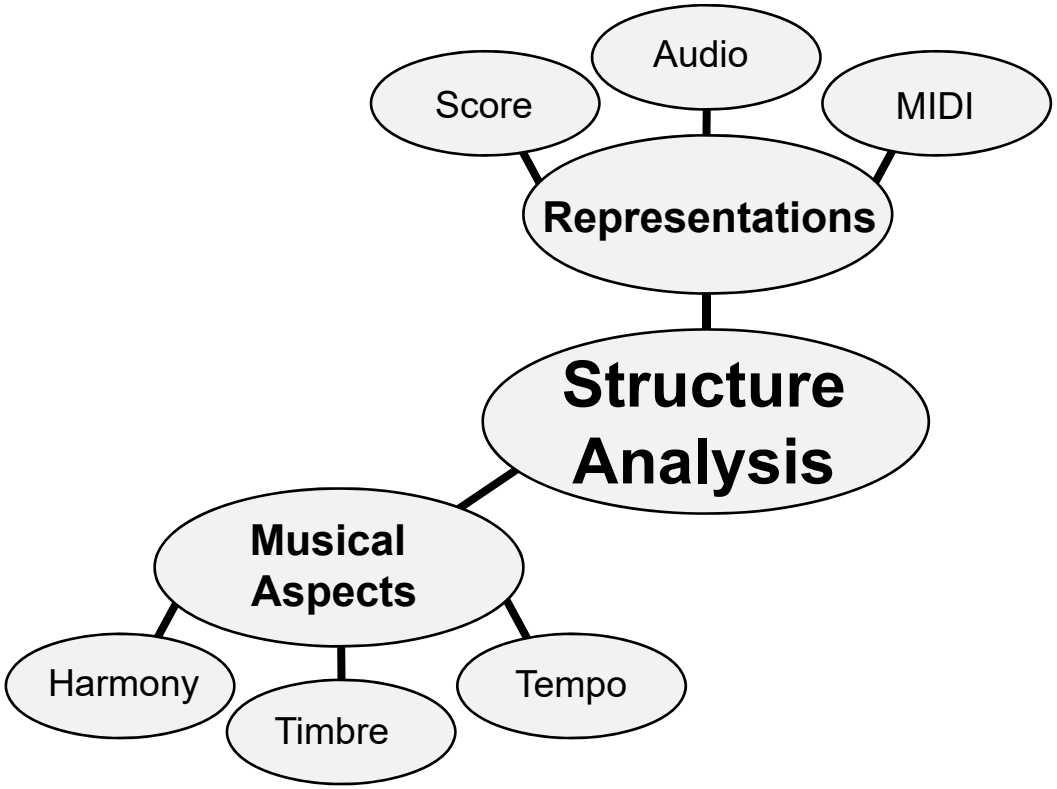
Conclusions



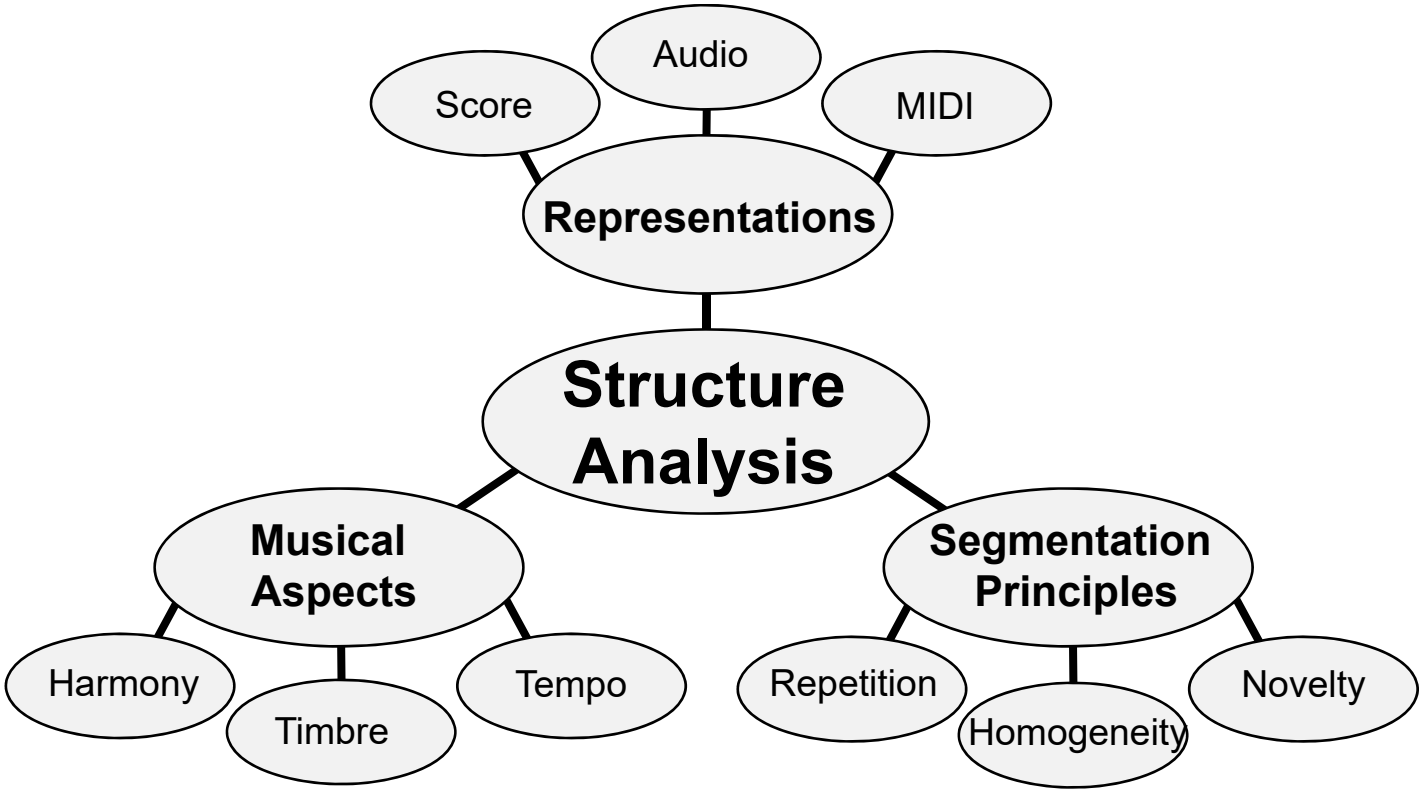
Conclusions



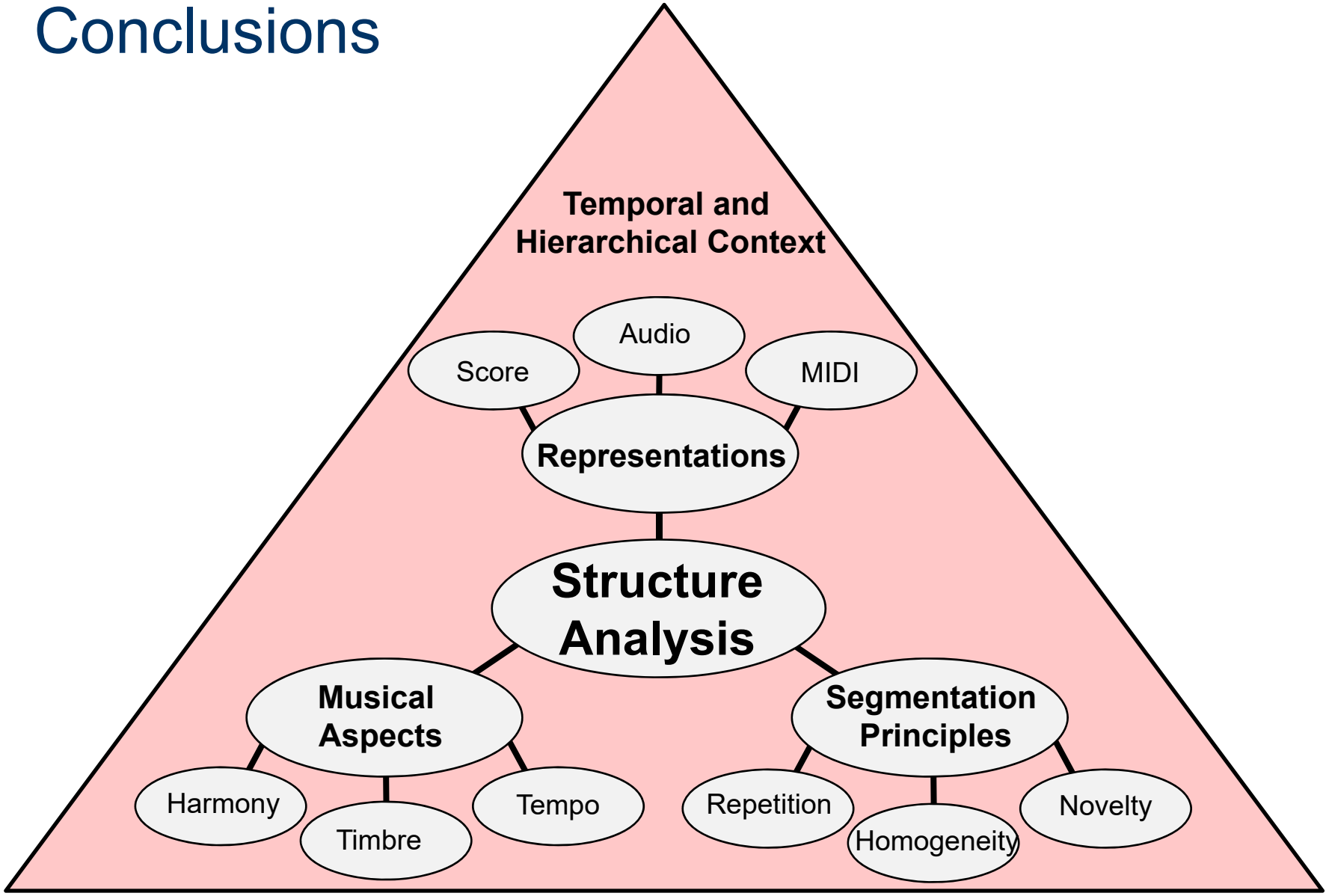
Conclusions



Conclusions



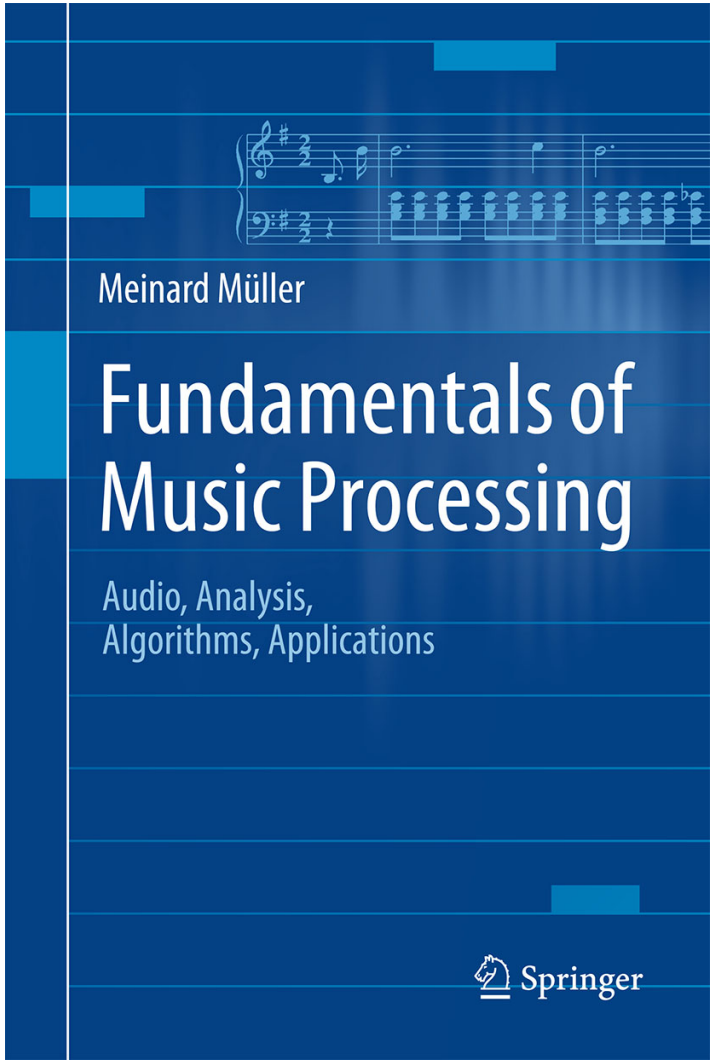
Conclusions



Links

- SM Toolbox (MATLAB)
  http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/
- MSAF: Music Structure Analysis Framework (Python)
  https://github.com/urinieto/msaf
- SALAMI Annotation Data
  http://ddmal.music.mcgill.ca/research/salami/annotations
- LibROSA (Python)
  https://librosa.github.io/librosa/
- Evaluation: mir_eval (Python)
  https://craffel.github.io/mir_eval/
- Deep Learning: Boundary Detection
  Jan Schlüter (PhD thesis)

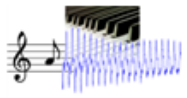
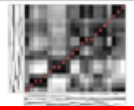
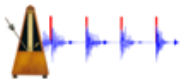
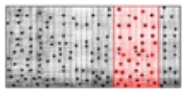
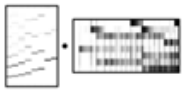
Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Book: Fundamentals of Music Processing

Chapter		Music Processing Scenario
1		Music Represenations
2		Fourier Analysis of Signals
3		Music Synchronization
4		Music Structure Analysis
5		Chord Recognition
6		Tempo and Beat Tracking
7		Content-Based Audio Retrieval
8		Musically Informed Audio Decomposition

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de